

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

CRÉATION, PARTAGE ET TRANSFERT D'ENSEMBLES DE
DONNÉES TERMINOLOGIQUES BASÉS SUR SKOS

MÉMOIRE DE RECHERCHE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA
MAÎTRISE EN INFORMATIQUE DE GESTION

PAR
KAISER ZOGLAMI

AOÛT 2011

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Au terme de ce mémoire, je tiens à adresser mes remerciements et à témoigner toute ma reconnaissance aux personnes qui ont participé de près ou de loin à la réalisation de cette recherche.

Mes remerciements vont tout d'abord à ma directrice Brigitte Kerhervé, professeure à l'Université du Québec à Montréal, pour m'avoir donné la chance de travailler sur ce projet de recherche et surtout pour m'avoir soutenu tout au long de ce projet par sa disponibilité, sa patience ainsi que ses nombreuses idées, ses remarques et son soutien.

Je remercie par ailleurs mon codirecteur Olivier Gerbé, professeur à l'Université de HEC à Montréal, pour son aide au cours du projet. Je le remercie en outre pour ses remarques et son aide dans la réalisation de ce mémoire.

Je tiens également à remercier les professeur(e)s que j'ai eu à rencontrer au cours de ma maîtrise à l'UQAM pour la qualité de leur enseignement, tout comme je tiens à remercier l'ensemble du personnel du département d'informatique de l'UQAM pour leur conseil, leur hospitalité et leur travail dévoué.

Finalement et non les moindres, je tiens à remercier mes parents qui m'ont toujours aidé et soutenu, ce mémoire leur est dédié.

TABLE DES MATIÈRES

LISTES DES FIGURES	ix
LISTES DES TABLEAUX	xiii
RÉSUMÉ	xiv
INTRODUCTION	1
CHAPITRE I	
ÉTAT DE L'ART	7
1.1. Introduction.....	7
1.2. Les systèmes d'organisation des connaissances	7
1.2.1 Les Taxonomies	8
1.2.2 Les glossaires	8
1.2.3 Les thésaurus.....	9
1.2.4 Les ontologies	11
1.3. Les standards.....	13
1.3.1 Le standard ISO 2788 & ISO 5964	13
1.3.2 British Standard BS 8723	14
1.3.3 SKOS « Simple knowledge organisation system »	14
1.4. Les outils terminologiques	15
1.4.1 Superthes	15
1.4.2 ThManager	16
1.4.3 SKOS Editor.....	17
1.4.4 ONKI SKOS.....	18
1.5. Conclusion	19
CHAPITRE II	
OBJECTIFS ET MÉTHODOLOGIE DE RECHERCHE	21
2.1. Introduction.....	21
2.2. Présentation de la DSRM ou « Design Science Research Methodology »	22
2.2.1 Activité 1 : Identification du problème.	23
2.2.2 Activité 2 : Définir les objectifs de la solution.....	23
2.2.3 Activité 3 : Conception et développement	23
2.2.4 Activité 4: Démonstration	23
2.2.5 Activité 5: Évaluation.....	24

2.2.6 Activité 6 : Communication « Archer, Hevner et al ».....	24
2.2.7 Conclusion.....	24
2.3. Projet de recherche.....	26
2.3.1 Activité 1: Identification du problème	26
2.3.2 Activité 2: Définition des objectifs de la solution	26
2.3.3 Activité 3: Conception et développement	27
2.3.4 Activité 4 - 5: Démonstration & Évaluation	27
2.3.5 Activité 6: Communication	27
2.4. Conclusion	27
CHAPITRE III	
MÉTA-MODÈLE DE SKOS.....	29
3.1. Introduction.....	29
3.2. Le standard SKOS.....	29
3.2.1 Historique.....	29
3.2.2 Les éléments de SKOS	30
3.2.2.1 Le concept SKOS.....	30
3.2.2.2 Les Labels SKOS	31
3.2.2.3 Les schémas de concepts.....	32
3.2.2.4 Les Collections de concepts.....	33
3.2.2.5 La documentation.....	33
3.2.2.6 Les relations sémantiques dans SKOS	35
3.2.2.7 SKOS eXtension Label.....	36
3.2.3 Évolution du standard.....	36
3.3. Modélisation conceptuelle du standard.....	37
3.3.2 Conclusion.....	41
3.4. Schéma de la BD supportant SKOS.....	44
3.5. Validation du modèle conceptuel de SKOS.....	45
3.5.1 Schéma de concepts :	45
3.5.2 Labels lexicaux.....	46
3.5.3 Documentation	48
3.5.4 Collection de concepts	48
3.5.5 Relations sémantiques.....	50
3.5.5.2 Relations de mapping.....	53

3.5.6 Étude de cas de l'observatoire international des coopératives de services financiers	55
3.6. Conclusion	60
CHAPITRE IV	
LE SYSTÈME LOGICIEL	61
4.1. Introduction.....	61
4.2. Présentation de l'outil logiciel	61
4.2.1 Architecture.....	61
4.2.2 Fonctionnalités du logiciel	63
4.2.3 Environnement et langages de développement	64
4.2.3.1 MySQL	65
4.2.3.2 L'EDI « Netbeans 6.8 »	65
4.2.3.3 Java Server Pages ou JSP	65
4.2.3.4 XML.....	66
4.3. Construction de la base de données	66
4.3.1 Le logiciel « ROBOT »	67
4.3.2 Insertion de la base de données SKOS	68
4.4. Développement de SKOS TOOL.....	72
4.4.1 Les fonctionnalités du système logiciel.....	73
4.4.1.1 Gestion des glossaires	73
4.4.1.2 Gestion des thésaurus.....	76
4.4.1.3 Gestion des glossaires.....	77
4.4.1.4 Exporter des ensembles terminologiques basés sur SKOS	78
4.4.1.5 Élaboration de KOS	78
4.4.1.6 Gestion des concepts SKOS.....	80
4.4.1.7 Gestion des langues.....	80
4.5. Mécanisme de « Mapping » entre les KOS et SKOS.....	80
4.5.1 Règles de correspondance pour les glossaires.....	81
4.5.1.2 Le glossaire sous SKOS.....	82
4.5.1.3 Le terme d'un glossaire en SKOS.....	83
4.5.1.4 La définition en SKOS.....	84
4.5.1.5 La langue sous SKOS	85
4.5.1.6 Conclusion	85
4.5.2 Règles de correspondance pour les thésaurus	86

4.5.2.2 Thésaurus sous SKOS.....	87
4.5.2.3 Terme d'un thésaurus en SKOS.....	88
4.5.2.4 Les relations d'un thésaurus en SKOS.....	89
4.5.2.5 Conclusion	92
4.6. Conclusion	93
CHAPITRE V	
EXPÉRIMENTATION.....	95
5.1. Introduction.....	95
5.2. Création et manipulation d'un glossaire	95
5.2.1 Présentation du cas d'étude et du scénario de test.....	95
5.2.2 Expérimentation	96
5.2.2.1 Scénario de l'expérimentation	96
5.2.2.2 Exécution du scénario	97
5.2.3 Conclusion et limites.....	102
5.3. Création et manipulation d'un thésaurus	102
5.3.1 Présentation du cas d'étude.....	102
5.3.2 Expérimentation	103
5.3.2.1 Scénario de l'expérimentation	103
5.3.2.2 Exécution du scénario	104
5.3.3 Conclusion et limites.....	110
5.4. Création et manipulation d'un système de gestion des connaissances « KOS ».....	110
5.4.1 Expérimentation	111
5.4.1.1 Scénario d'expérimentation	111
5.4.1.2 Exécution du scénario	111
5.4.1.3 Conclusion et limites.....	115
5.5. Conclusion	116
CONCLUSION.....	117

LISTE DES FIGURES

Figure	Page
1.1 - Partie d'un glossaire génomique	9
1.2 – Un exemple de Thésaurus	11
1.3 - Exemple d'une ontologie	13
2.1 - Modèle de processus du « Design Science Research Method »	25
3.1 – Description du fonctionnement de SKOSXL.....	36
3.2 – Les entités du modèle conceptuel de SKOS	38
3.3 - Transformation de l'association « skos:member ».....	39
3.4 - Les relations du modèle conceptuel.....	41
3.5 - Modèle conceptuel de SKOS et ses relations	43
3.6 - Schéma de la base de données SKOS.....	44
3.7 - Exemple consistant d'un concept schème	46
3.8 - Diagramme d'objet de l'exemple 5.....	46
3.9 - Exemple consistant d'un concept labélisé.....	47
3.10 - Diagramme d'objet de l'exemple 10.....	47
3.11 - Exemple consistant de la propriété "Note" de SKOS	48
3.12 - Diagramme d'objet correspondant au « skos:note »	48
3.13 - Exemple consistant d'une collection de concepts	49
3.14 - Diagramme d'objet relatif à l'exemple 40	49
3.15 - Exemple consistant d'une collection ordonnée de concepts	49
3.16 - Diagramme d'objet relatif à la sous-propriété « skos:orderedCollection ».....	50
3.17 - Exemple d'une relation sémantique inconsistante	50
3.18 - Diagramme d'objet de l'exemple 32.....	51
3.19 - Exemple consistant de relations sémantiques.....	51
3.20 - Diagramme d'objet relatif au « skos:broader » de l'exemple 35	52
3.21 - Diagramme d'objet relatif au « skos:broaderTransitive » de l'exemple 35	52
3.22 - Exemple consistant d'une relation de mapping	53
3.23 - Diagramme d'objet relatif à la sous-propriété « skos:relatedMatch »	54
3.24 - Exemple consistant d'une relation de mapping.....	54
3.25 - Diagramme d'objet relatif à la sous-propriété « skos:exactMatch »	55

3.26 - Proposition d'utilisation du label SKOS	57
3.27 – Exemple du « Banco Credicoop » illustrant la première solution.....	57
3.28 - Exemple du « BVR » illustrant la première solution.....	58
3.29 - Proposition de représentation en utilisant SKOS-XL.....	58
3.30 - Exemple du « Banco Credicoop » illustrant la deuxième solution.....	59
3.31 - Exemple du « BVR » illustrant la deuxième solution	59
4.1 - Architecture physique de l'outil de gestion des ensembles de données terminologiques basée sur SKOS.....	62
4.2 - Diagramme de cas d'utilisation de « SKOS TOOL »	64
4.3 - Diagramme de classe d'un modèle « entité-association ».....	67
4.4 - Vue présentée à l'utilisateur de la base de données.	67
4.5 - Représentation XML de la table « concept » de l'entité Concept du modèle conceptuel de SKOS	69
4.6 - Représentation XML de la relation entre « concept - inscheme » et celle entre « conceptscheme - inscheme ».....	70
4.7 – Modèle entité-association de SKOS	71
4.8 - Représentation conceptuel de tables métiers correspondant à la fonctionnalité de gestion des KOS.....	72
4.9 - Création et édition d'un glossaire.....	74
4.10 - Création d'un nouveau terme d'un glossaire	75
4.11 - Affichage détaillé d'un terme d'un glossaire.....	76
4.12 - Affichage d'un terme du thésaurus « Law Congress ».....	77
4.13 - Contenu du thésaurus « Law Congress » après exportation	78
4.14 - Création d'un nouveau système de gestion des connaissances	79
4.15 - Représentation UML d'un glossaire.....	82
4.16 - Modèle conceptuel des thésaurus	87
5.1 - Glossaire de la Banque du Canada	96
5.2 - Processus de création du glossaire « Banque du Canada ».....	97
5.3 – Processus d'insertion d'un nouveau terme dans le glossaire de « Banque du Canada »	98
5.4 - Une partie de la fonction d'exécution du script sur un glossaire	99
5.5 - Résultats de l'exécution du script d'expérimentation du mécanisme de « mapping » sur un glossaire	100
5.6 - Processus de création d'un thésaurus	104
5.7 - Insertion du terme « Crédit ».....	105

5.8 - Affichage des propriétés du terme « Crédit » avant et après ajout des relations	106
5.9 - Résultats de l'exécution du script d'expérimentation du mécanisme de « mapping » sur le nouveau thésaurus	107
5.10 - Processus de création d'un nouveau KOS	112
5.11 - Résultats d'exécution du script de vérification du mécanisme de « mapping » du nouveau KOS	113
5.12 - Processus de création d'un « Glosaurus »	114
5.13 - Interface d'insertion et d'affichage du terme « activité économique » dans un « Glosaurus »	114
5.14 - Résultats d'exécution du script d'expérimentation du mécanisme de mapping suite à une insertion du terme « cycle économique »	115

LISTE DES TABLEAUX

Tableau	Page
3.1 - Echantillon des acronymes de l'observatoire international des coopératives	56
4.1 - Règle de correspondance pour un glossaire.....	83
4.2 - Règles de correspondance pour un terme et son libellé.....	84
4.3 - Règle de correspondance pour une définition d'un glossaire.....	85
4.4 - Règle de correspondance pour la langue	85
4.5 - Résumé des règles de correspondance.....	86
4.6 - Règle de correspondance d'un thésaurus et des relations en SKOS	88
4.7 - Règles de correspondance pour un terme et son libellé.....	89
4.8 - Règles de correspondance entre les relations d'un thésaurus et SKOS	91
4.9 - Règles de correspondance entre les relations synonymiques d'un thésaurus et SKOS.....	92
4.10 - Résumé des règles de correspondances	93
5.1 - Evaluation du contenu de la BD	101
5.2 - Thésaurus de l'Activité Gouvernementale	103
5.3 - Synthèse de l'analyse entre les résultats d'exécution du script et des résultats théoriques.....	109

—

--

—

--

—

—

—

RÉSUMÉ

Depuis quelques années, la recherche sur la normalisation des ensembles de données terminologiques a été une des plus grandes préoccupations d'un groupe de travail du World Wide Web Consortium. Ce travail de recherche a permis de développer un nouveau standard SKOS qui a été ensuite émis en 2009 en tant que recommandation du W3C. Le travail de recherche que nous entreprenons dans ce mémoire concerne ce standard. Notre étude consiste à réaliser à partir de cette recommandation un modèle conceptuel fiable et cohérent reprenant les principaux éléments de SKOS et de son extension. Elle consiste aussi à prototyper un outil reposant sur ce modèle qui permettrait, de gérer, de manipuler et de partager des ensembles de données terminologiques basés sur SKOS. Pour ce faire, une méthodologie de recherche bien précise a été adoptée et diverses recherches concernant le standard SKOS et le type d'ensemble de données terminologiques ont été réalisées. A partir de ces recherches, nous avons proposé un modèle conceptuel de SKOS et nous avons conçu et prototypé l'outil le supportant, qui a été par la suite testé à travers des cas d'utilisation réels. Ces tests ont permis d'évaluer l'outil et de démontrer sa capacité à manipuler des données normalisées en SKOS en toute simplicité.

Mots clés : Gestion de connaissances, SKOS, RDF, gestion des KOS, ensembles de données terminologiques basés sur SKOS, exportation SKOS, glossaires, thésaurus.

INTRODUCTION

Le mot a toujours occupé une place importante dans notre vie quotidienne, que ce soit pour s'exprimer ou pour communiquer. Il est un élément central qui permet aux différents intervenants de se comprendre. Parfois les mots sont insuffisants pour permettre une compréhension totale. Vu la richesse de notre langage et les différentes langues qui existent, plusieurs mots possèdent différentes significations et il arrive, par conséquent, que des personnes ne comprennent pas des mots ou des termes particuliers liés à un vocabulaire donné, employé dans un environnement spécifique. Un terme propre à la cuisine ne pourrait pas être interprété par un novice de la même façon qu'un expert, de même qu'un employé fraîchement recruté dans une entreprise pourrait ne pas comprendre certains vocabulaires métiers employés pour décrire, par exemple, un processus d'affaire. Ainsi, pour qu'ils puissent mener à bien leur mission, ils doivent préalablement convenir ou comprendre la signification des termes spécifiques à leur environnement et ainsi éviter les ambiguïtés relatives au sens. Pour réussir une recette culinaire, les débutants doivent comprendre, tout d'abord, les termes employés dans un livre de cuisine, tout comme un employé doit comprendre le sens des termes décrivant le processus qu'il doit exécuter.

Ainsi, tel deux personnes qui parlent deux langues non compréhensibles par l'un et l'autre, se mettre d'accord sur la terminologie est essentiel pour faciliter non seulement les échanges mais aussi certaines opérations telles que la recherche d'information.

1.1. PROBLEMATIQUE

Avec l'explosion de la masse d'informations sur le Web, la compréhension du contenu informationnel est devenue difficile. En effet, les informations déployées sur le web sont, majoritairement, non structurées. Cette non-structuration des données complique leur interprétation ainsi que leur échange entre les applications disposées sur des machines.

L'avènement du Web sémantique (T. Berners-Lee et al., 2001) était destiné à résoudre le problème de non-structuration des informations. Il visait à rendre le contenu des

ressources du Web utilisable en formalisant les données, grâce à des ensembles terminologiques. Ces derniers ont pu résoudre des problèmes relatifs au partage, à l'échange et à la recherche d'informations (A. Miles et al., 2009). En effet, les ensembles terminologiques permettent de fournir une description consentie sur des termes spécifiques d'un domaine particulier et dans certains systèmes d'organisation des terminologies, il est possible de définir des relations entre les différents termes, à travers par exemple, des relations d'équivalence. Il devient, alors, facile pour un agent d'obtenir l'information souhaitée et ce, en étendant, par exemple, automatiquement sa recherche à des termes équivalents ou encore de partager des informations qui seront correctement interprétées par ses utilisateurs. La représentation des données et des termes par des ensembles terminologiques favorise, dès lors la collaboration des différentes communautés.

Toutefois, bien qu'ils servent à structurer des données, les systèmes d'organisation des connaissances, tels que les taxonomies (M. Zacklad, 2007), les ontologies (R. Gruber, 1994) et les thésaurus (M. Zacklad, 2007) demeurent exploitables par quelques outils. Certains travaux ont pu aboutir à la production d'outils de gestion des ensembles terminologiques tels que « THESMain » (R. Lega et al., 1998), « SuperThes » (Batschi et al., 2000). Certes, ils permettent de créer des thésaurus, de les partager et de rechercher des données qui y sont contenues, cependant, le handicap majeur de ces outils est qu'ils ne traitent que des thésaurus qui sont une forme parmi tant d'autres d'ensembles terminologiques. Par ailleurs, la construction d'un thésaurus peut s'effectuer de différentes manières, soit en suivant un standard ou tout simplement selon les besoins des utilisateurs. Cette différence de construction aboutit à un écart entre la structure des deux types. Cette diversité au niveau des propriétés des ensembles terminologiques empêche l'élaboration d'un outil apte à les gérer efficacement.

C'est dans cette perspective qu'un groupe de travail (SWDWG) du W3C « World Wide Web Consortium » (2001) s'est penché sur l'étude des ensembles terminologiques. Cette étude a permis de déterminer un dénominateur commun aux différents ensembles terminologiques et a permis la réalisation d'un modèle de données commun aux différents

systèmes d'organisation des terminologies. Celui-ci, connu sous le nom de SKOS¹ « Simple Knowledge Organization System » (A. Miles et al., 2009, A. Miles et al., 2006) est un méta-modèle qui normalise la représentation des ensembles terminologiques grâce à une famille de langages formels. Il permet de représenter n'importe quel type de systèmes d'organisation des terminologies et de les rendre interoperables.

Depuis l'annonce de ce langage de standardisation, quelques chercheurs se sont penchés sur la réalisation d'outils permettant la création, le partage et le transfert ainsi que la recherche des différents ensembles terminologiques basés sur SKOS. *J. Lacasta et al.* (2007), sont parvenus, à travers leurs travaux, à développer l'outil « *ThManager* ». Ce dernier a été créé dans le but de gérer des thésaurus normalisés sous le format SKOS. Il offre, en plus des fonctionnalités de visualisation du contenu des thésaurus, de son édition et de sa suppression, d'intégrer des thésaurus et d'autres vocabulaires contrôlés dans des applications qui requièrent leur utilisation. Le travail de *J. Tuominen et al* (2009) n'est pas sans reste. Il a permis de réaliser un outil dénommé « *ONKI SKOS* ». Cet outil, facile d'utilisation, permet de publier, d'intégrer et d'utiliser des thésaurus formalisés en SKOS via le Web. Les différentes phases de test auxquelles il a été confronté, ont valu à ONKI d'être le canal de publication officiel du grand thésaurus finlandais « *YSA* ». *S. Jupp et al.* (2008) proposent eux aussi un éditeur pour SKOS désigné par « *SKOS Editor* ». Ce dernier a été développé afin d'être intégré dans un outil ontologique (Protégé 4) et ainsi permettre l'usage de SKOS depuis cette application.

Tous ces outils, bien qu'ils aident efficacement à la gestion de thésaurus, possèdent certaines limites. Tout d'abord, certaines applications ne sont pas accessibles à toutes les catégories d'utilisateurs. Par exemple « *ONKI SKOS* » requiert au préalable la conversion des thésaurus sous le format SKOS ou RDFS. Or, cette opération, à défaut de n'être pas réalisée par l'application, nécessite l'intervention d'un expert. De plus, certains de ces outils, notamment « *SKOS editor* » ont été développés dans le but d'intégrer la gestion des thésaurus dans des applications telles que Protégé et ce, en définissant des règles d'utilisation qui en restreignent les possibilités. Par ailleurs, la majorité de ces outils ont été élaborés pour un ensemble de terminologies spécifiques, les thésaurus. Or

¹ www.w3.org/2004/02/skos/

les ensembles terminologiques n'englobent pas seulement cette forme de système d'organisation des connaissances. Finalement, l'ensemble de ces outils ne tiennent pas compte des évolutions du modèle SKOS. Depuis sa création, ce méta-modèle n'a jamais cessé d'évoluer.

Les travaux initiés récemment par les professeurs Brigitte Kerhervé et Olivier Gerbé ont pour objectif de proposer une approche orientée modèles permettant de gérer facilement des ensembles de données terminologiques. Cette approche est basée sur la définition et la transformation de modèles conceptuels. Les deux chercheurs travaillent actuellement à la modélisation conceptuelle de différents ensembles de données terminologiques pour pouvoir par la suite travailler à la transformation de ces modèles vers le modèle conceptuel de SKOS et développer un outil générique.

Le défi principal d'un tel outil est d'effectuer les correspondances entre les différentes propriétés d'un ensemble terminologique, tel un thésaurus, avec celles du modèle conceptuel de SKOS. Les relations entre les termes d'un thésaurus ou d'un schéma de classification peuvent changer lors du processus de mise en correspondance entre le modèle d'un ensemble terminologique et le modèle conceptuel SKOS.

Ainsi, la problématique qui se pose concerne la conception et le développement d'un outil basé sur la transformation de modèles conceptuels permettant de créer, de partager et de transférer des ensembles terminologiques basés sur SKOS. Plus explicitement la question que nous nous posons est la suivante : Quelles sont les approches et les stratégies qui peuvent être proposées pour réaliser l'intégration et le partage d'ensembles de données terminologiques basés sur des modèles conceptuels différents ? Comment implémenter de telles approches dans l'environnement SKOS ?

Afin de résoudre cette problématique et de réaliser nos objectifs, nous commençons dans le chapitre suivant de ce mémoire par l'étude de l'existant en nous intéressant aux travaux réalisés par des scientifiques du domaine et aux différentes approches que nous pourrions utiliser. Dans le troisième chapitre, nous explicitons nos objectifs de recherche et nous présentons la méthodologie employée pour aboutir à notre fin. Une fois celle-ci exposée, nous passons, au cours du chapitre quatre, à la présentation du méta-modèle SKOS, la base de données qui le supporte ainsi qu'à sa validation. Ensuite, nous entamons le

chapitre « Système logiciel » qui nous présente l'architecture physique de l'outil à développer, ses fonctionnalités, sa conception et les outils employés pour sa réalisation. Finalement, nous testons et nous validons notre logiciel lors du sixième chapitre pour finir dans le dernier chapitre par discuter les résultats, les limites et l'atteinte de nos objectifs.

CHAPITRE II

ÉTAT DE L'ART

2.1. INTRODUCTION

Depuis l'adoption du standard SKOS par le W3C, plusieurs outils proposent de gérer différents systèmes d'organisation des connaissances. Ce chapitre a pour but de présenter, dans un premier temps, une liste exhaustive de ces outils ainsi que les multiples systèmes d'organisation des connaissances et les standards existant qui permettent de gérer les KOS².

A travers la liste des systèmes d'organisation des connaissances, nous essayons d'apprendre sur leurs différentes propriétés pour mieux appréhender, d'une part, leur comportement et d'autre part, pour comprendre les standards qui leur sont appropriés. Finalement, nous présentons le standard SKOS et ses propriétés ainsi que les outils qui se basent sur ce standard pour gérer les KOS.

2.2. LES SYSTEMES D'ORGANISATION DES CONNAISSANCES

Les systèmes d'organisation des connaissances visent à regrouper et à organiser les connaissances selon leurs domaines d'appartenance tels que l'informatique, la biologie, etc. Ils ont pour objectif de faciliter la collaboration entre différentes entités, en déterminant un vocabulaire commun.

² KOS ou « Knowledge Organisation System » équivaut aux systèmes d'organisation des connaissances.

Ces ensembles se structurent autour du mot. Étant donné la richesse du langage naturel, un mot possède une multitude de sens, de définitions, de synonymes. Ainsi, les ensembles terminologiques servent à enrichir ce mot, en lui associant des connaissances, comme par exemple, ses synonymes.

Afin de regrouper les différents mots ou termes d'un domaine, différents ensembles terminologiques peuvent être utilisés.

2.2.1 Les taxonomies

Les taxonomies sont une forme de système d'organisation des connaissances qui regroupe les schémas de classification et les « subjects heading ». Elles ont été, initialement, employées pour étudier et grouper les organismes vivants en familles (Peter H. Raven et al, 1971). Ensuite, elles ont été reprises par l'informatique pour ordonner et classer des termes sous la forme d'arborescence. Généralement, une taxonomie tend à organiser les termes en recourant à des relations de généralité-spécificité. Étant donné qu'une taxonomie est représentée sous forme d'arbre (Godfray et H Charles J, 2002), elle possède alors une racine, ainsi qu'une multitude de nœuds connus sous le nom de taxons, chacun représentant un terme. Ce dernier peut avoir plusieurs parents. Par exemple, un « vélo » peut appartenir à la fois au parent « transport » et au parent « sport ». Cette relation hiérarchique sous-entend aussi que les contraintes appliquées sur un nœud parent sont applicables aux nœuds fils.

2.2.2 Les glossaires

Le mot glossaire est apparu au 16^{ème} siècle, il désignait anciennement un dictionnaire expliquant certains mots obscurs d'une langue par d'autres termes de la même langue. En d'autres mots, un glossaire est considéré comme un dictionnaire spécialisé. Il réunit tous les termes spécifiques, connus ou étrangers, à un domaine précis ainsi que leur définition. Un glossaire possède généralement un index et ses termes sont ordonnés alphabétiquement pour faciliter la recherche.

Un glossaire peut contenir un terme utilisé dans d'autres glossaires dans lesquels il possède une définition entièrement dissemblable. Nous pouvons parler du mot « format » qui est à la fois utilisé dans un glossaire d'art et dans celui de l'informatique. L'un définit

le format comme étant les dimensions d'un papier ou d'une toile alors que l'autre, le décrit comme étant le type de fichier.

Ci-dessous, nous présentons un extrait d'un glossaire du laboratoire de recherche « Amabiotics » et utilisé par son fondateur lors de ses cours à l'École Normale Supérieure en France.

Glossaire Génomique

Gène

Le gène est l'atome fonctionnel de l'hérédité. Il définit un produit (*ARN* ou *protéine*) et des éléments de contrôle. La définition du gène a beaucoup varié au cours de l'histoire (la définition initiale due aux travaux de Mendel était opérationnelle, elle ne faisait pas référence à une structure physique). Il est souvent facile de définir la partie codante d'un gène (c'est la portion du chromosome qui code son produit, une *protéine* le plus souvent). Il est beaucoup plus difficile d'en définir les bords. Il s'ensuit de nombreuses discussions, souvent animées ou même violentes. Il ne s'agit pas simplement de querelles d'experts, mais d'un problème crucial, si l'on veut mettre en œuvre efficacement des bases de données gérant les connaissances biologiques (il faut être précis pour pouvoir manipuler des concepts à l'ordinateur)

Génome

L'ensemble des *gènes* d'un organisme au sens physique de son *ADN*.

Génotype

L'ensemble des *gènes* d'un organisme par opposition à la façon dont il s'exprime dans un individu, son *phénotype*

Figure 2.1 - Partie d'un glossaire génomique (source: <http://www.normalesup.org/~adanchin/populus/glossaire.html>)

2.2.3 Les thésaurus

Tout comme les glossaires, les thésaurus contiennent un « ensemble de termes sélectionnés à partir du langage naturel » (Austin et Waters, 1980), relatifs à un domaine, servant à l'indexation des documents. Ils contiennent des mots descripteurs et non descripteurs, c'est-à-dire des termes qui ne peuvent pas être utilisés pour indexer, ainsi qu'un sous ensemble de définitions et de notes (Zacklad, 2007).

L'organisation internationale de la standardisation définit les thésaurus comme étant « le vocabulaire d'un langage d'indexation contrôlé organisé formellement de façon à expliciter les relations à priori entre les notions » (ISO 5964, 1985) Les thésaurus

cherchent donc à identifier les relations entre les différents termes, en définissant des relations d'équivalences, hiérarchiques et associatives, pour faciliter l'interrogation ultérieurement.

Les relations hiérarchiques sont les bases de la hiérarchie des thésaurus. Elles relient **un terme générique**, identifié par **TG**, à **un terme spécifique** « **TS** » à l'intérieur du champ sémantique du terme générique.

Les relations d'équivalences sont établies entre un terme descripteur et non-descripteur, ce dernier est nommé terme équivalent. Un **terme équivalent** « **EP** » (Employé Pour), est un terme équivalent à un terme spécifique mais présenté comme optionnel dans l'emploi du thésaurus.

Les relations associatives ne s'appliquent uniquement qu'à des termes descripteurs, plus précisément à des **termes associés** « **TA** ». Elles permettent de les enrichir sémantiquement en établissant, en outre, un lien avec d'autres termes d'un domaine connexe. Cette relation permet d'élargir la recherche de termes sans pour autant faire appel aux termes génériques.

-- L'élaboration d'un thésaurus peut s'effectuer soit manuellement en recourant à l'intelligence d'un agent humain, soit automatiquement par l'intermédiaire de logiciels de construction de thésaurus. L'élaboration automatique d'un thésaurus fait intervenir principalement l'intelligence artificielle pour déterminer la fréquence d'un mot clé et ainsi le définir comme index. —

Pour réaliser un thésaurus, diverses règles régissent leur contenu. L'organisme ISO « International Organization for Standardization » a défini deux principales normes :

- ISO 2788 (1986) pour les thésaurus monolingues.
- ISO 5964 (1985) pour les thésaurus multilingues.

Dans la suite, nous présentons un extrait d'un thésaurus relatif à la biologie, où l'on peut observer les relations de hiérarchie existant entre les termes.

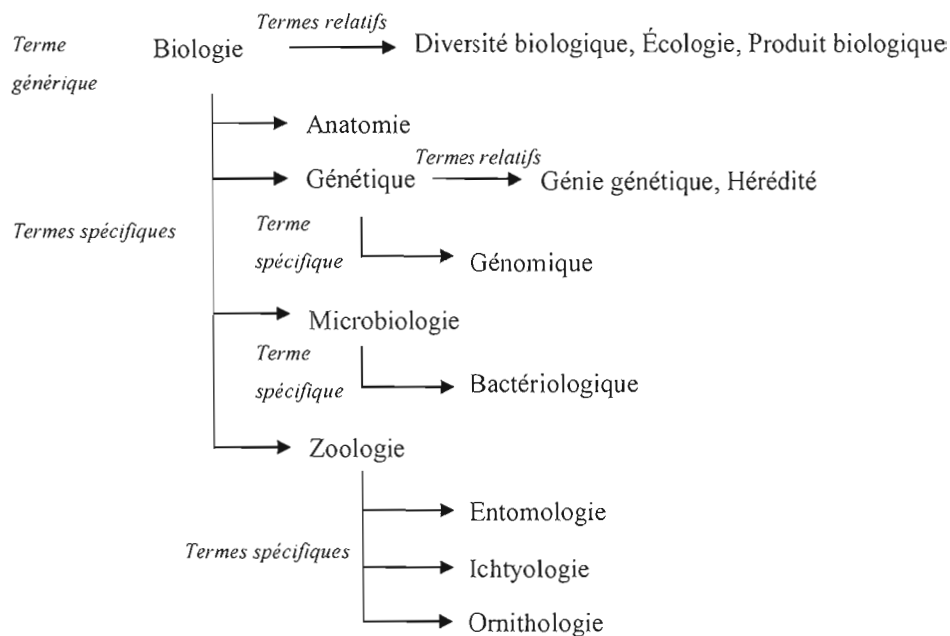


Figure 2.2 – Un exemple de Thésaurus (source : <http://www.thésaurus.gc.ca/cgi-bin/mtmcgi80.dll?s=1&n=1&t=3&l=1&k=1&y=0&w=Biologie>)

2.2.4 Les ontologies

L'ontologie a été longtemps employée en philosophie pour désigner l'étude des propriétés de l'existant et a été, ensuite, reprise par les sciences de l'informatique afin de désigner un modèle de connaissances (Thomas Gruber et al., 2008).

Thomas Gruber(1994), une des notoriétés du domaine, définit l'ontologie comme étant « une spécification conceptuelle d'un domaine, qui aide les humains et les programmes à partager des connaissances ». Cette définition s'appuie sur deux dimensions importantes. La conceptualisation et la spécification de cette conceptualisation. Autrement dit, une ontologie est un moyen de description formelle d'un domaine spécifique. Elle le représente sous la forme d'un ensemble structuré de termes et de notions abstraites dites « concepts ».

Les concepts ainsi que les relations et les axiomes forment les principaux composants d'une ontologie (T. Gruber, 1993). Un concept représente une ou plusieurs entités d'un domaine. On en distingue principalement deux sortes.

- Les concepts primitifs : ce sont des concepts de premier niveau possédant une seule propriété nécessaire à leur appartenance au domaine mais ne permettant pas d'affirmer, qu'à chaque interprétation, il en fera toujours partie. Par exemple le concept « organe » pourrait être un organe humain tout comme il pourrait être celui d'un ordinateur. Il ne possède pas les propriétés suffisantes pour le distinguer d'autres types d'organes et pour permettre de l'associer spécifiquement au domaine de l'anatomie.
- Les concepts définis : ce sont des concepts pour lesquels une ou plusieurs propriétés nécessaires et suffisantes d'appartenance doivent lui être associées. Cela permet de distinguer, par exemple, tous les organes humains de ceux des ordinateurs tels que la propriété « organe intestin grêle » qui permet d'identifier l'organe comme étant un élément anatomique.

Une fois les concepts créés, on peut décrire les interactions entre eux grâce à des relations. Les concepts sont généralement reliés soit par des relations sémantiques, comme des relations d'équivalence ou d'associativité, soit par des relations d'appartenance.

Outre les concepts et les relations, l'ontologie repose sur des axiomes. Ces derniers servent à définir des contraintes sur les concepts. Les relations entre les concepts peuvent être considérées comme une sorte d'axiomes.

La figure qui suit décrit les différentes relations que nous pouvons trouver dans une ontologie représentative de l'anatomie d'une partie du corps humain et précisément l'intestin.

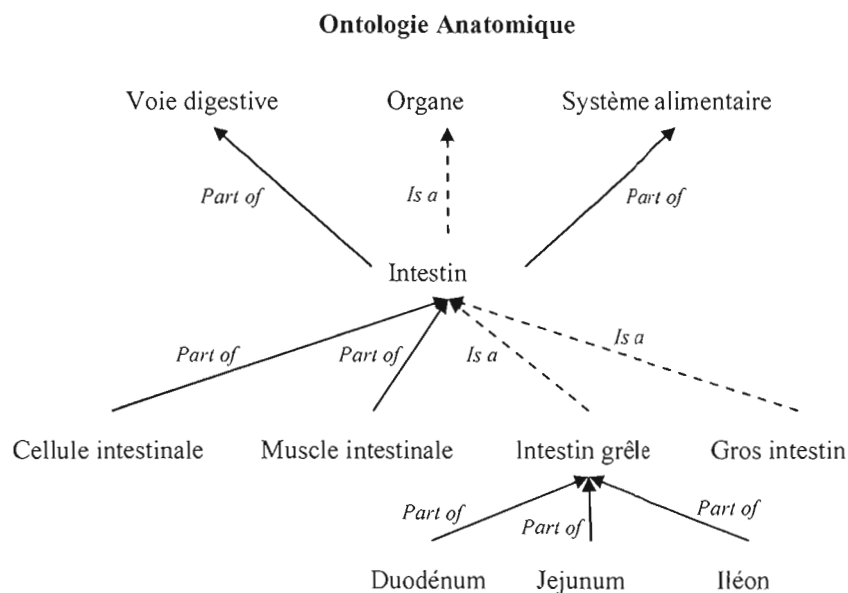


Figure 2.3 - Exemple d'une ontologie

(source: http://www.labunix.uqam.ca/~makarev/BIF7002/Rapport_HAB.htm)

2.3. LES STANDARDS

Afin de gérer de tels systèmes de gestion des connaissances, plusieurs organismes de standardisation se sont intéressés à la réalisation de normes. Parmi eux, nous citons principalement trois standards qui sont les plus connus et que nous explicitons dans la suite.

2.3.1 Le standard ISO 2788 & ISO 5964

ISO 5964, comme son titre l'indique « Guidelines for the establishment and development of multilingual thesauri », est une norme ISO pour la création et le développement de thésaurus multilingues. Il est destiné à être utilisé en communion avec la norme ISO 2788, qui régit la construction de thésaurus monolingue. En effet, ce standard tente de fournir aux gestionnaires de ces ensembles des conseils et donne une voie à suivre pour la construction des thésaurus. Il définit toutes les relations qui doivent être utilisées dans les thésaurus ainsi que les différentes propriétés dont il dispose ainsi que la syntaxe à employer pour qu'il puisse être bien interprété par d'autres utilisateurs.

2.3.2 British Standard BS 8723

Les normes ISO concernant les terminologies ont évolué grâce aux travaux réalisés par la British Standard à travers son projet BS 8723. Ce dernier, aussi connu sous le nom de « Structured vocabularies for information retrieval - Guide », offre à ses utilisateurs des orientations claires sur l'application de l'analyse des thésaurus, des informations complémentaires sur la gestion du développement et de la maintenance des thésaurus et des spécifications fonctionnelles lors de création de logiciels de gestion des thésaurus.

2.3.3 SKOS « Simple knowledge organisation system »

SKOS « Simple Knowledge Organization System », comme présenté ci-dessus, est un méta-modèle servant à structurer le contenu informationnel qui se trouve sur le Web et à exprimer des schémas de concepts tels que les thésaurus. SKOS a pu développer des standards de présentation de données qui ont permis de palier à des problèmes de non-interopérabilité des formats de représentation des thésaurus. Ainsi, une fois normalisés, ces derniers facilitent l'indexation, l'échange et l'intégration du contenu informationnel depuis n'importe quelle source en les rendant non-ambigus.

Afin de permettre une portabilité des ensembles terminologiques, le méta-modèle SKOS recourt au langage RDF (Ressource Description Framework). Ce dernier permet de représenter les données contenues dans les différents systèmes d'organisation des connaissances sous un format facilitant leur échange et leur exploitation par diverses applications informatiques.

Le méta-modèle SKOS repose sur des classes et des propriétés. Ainsi, pour représenter un ensemble de données terminologiques, SKOS recourt à un vocabulaire bien déterminé dont l'élément central est le « *Concept* ». SKOS utilise d'autres éléments que nous citons :

- Label
- Schéma de concepts
- Collection
- Documentation

– Relations sémantiques

Nous détaillerons ces derniers lors de la présentation du méta-modèle de SKOS dans le chapitre 4 de ce mémoire.

2.4. LES OUTILS TERMINOLOGIQUES

Afin de gérer et de manipuler ces ensembles de données terminologiques, divers chercheurs se sont impliqués dans le développement d'outils et systèmes de gestion des KOS. Dans ce qui suit, nous présentons les principaux outils que nous avons retenus.

2.4.1 Superthes

« SuperThes » (Batschi et al., 2000) est un système permettant de gérer et de maintenir les thésaurus. Il a été développé sur la base du « THESmain »³ (R. Lega et al., 1998) afin de profiter de l'expérience acquise par ce dernier.

Cet outil permet de construire et de maintenir, à la fois, des thésaurus monolingues et multilingues conformément à la norme ISO 2788 et 5964, et d'effectuer d'autres opérations dont la suppression, la copie de thésaurus et la visualisation de leur contenu soit par l'intermédiaire d'un tableau, soit à travers une vue hiérarchique.

« SuperThes » renferme une multitude de fonctionnalités autres que celles citées ci-dessus. En effet, en plus de visualiser, de créer, de supprimer et de modifier les thésaurus, cet outil permet d'ajouter et de mettre à jour toutes les données contenues dans un thésaurus. Il permet, par ailleurs, de sauvegarder différents types de données complexes telles que les images et les textes formatés. De plus, « SuperThes » fournit une interface pour les applications destinées à fonctionner avec cet outil afin de profiter de ses fonctionnalités. Il offre aussi l'importation et l'exportation de données vers des applications Microsoft, comme MS Access, et de lire les thésaurus formatés en SGML (dédié à l'agence européenne de l'environnement⁴).

³ Thesaurus Maintenance System « THESmain » a été conçu pour gérer et maintenir les deux thésaurus CDS et UDK (Umweltdatenkatalog) http://www.cedar.at/wrg_home

⁴ <http://www.eea.europa.eu/>

2.4.2 ThManager

Le « ThManager » (Javier Lacasta et al., 2007) est un outil qui a été développé pour gérer les thésaurus formalisés sous SKOS. Cet outil développé en Java, repose sur une architecture à trois couches. Chacune de ces couches joue un rôle important dans le fonctionnement de « ThManager ».

La première couche, le « Repository layer », permet le stockage des thésaurus dans des entrepôts. Aussi, permet-elle d'entreposer des métadonnées relatives aux thésaurus et ce, pour pouvoir les classer et les identifier facilement. La seconde couche « GUI »⁵ permet de visualiser, d'éditer, d'importer et d'exporter sous format SKOS les thésaurus. Cette couche offre aussi aux différentes applications qui veulent exploiter des thésaurus stockés dans le « ThManager » d'utiliser les diverses composants du GUI ou les méthodes fournies par la couche de persistance. Cette dernière constitue la troisième couche du « ThManager ». Elle représente la couche intermédiaire de cet outil. La « Persistence layer » est une API qui contient toutes les méthodes et fonctions permettant de gérer les thésaurus efficacement.

Outre cette architecture particulière, cet outil offre de bonnes performances dont une rapidité d'exécution et une gestion optimale des espaces de stockage. En effet, bien que les thésaurus soient volumineux, « ThManager » arrive à réduire considérablement leur taille et ce, en les transformant dans un format binaire. Par ailleurs, il permet d'inter-relier un ou plusieurs thésaurus provenant de différents domaines d'activité et d'enlever les ambiguïtés qu'il peut y avoir grâce à une méthode de désambiguïsation développée par Noguera-Iso et al. (2005).

Bien que permettant une très bonne gestion des thésaurus et des vocabulaires contrôlés, cet outil ne permet pas cependant de représenter des propriétés de SKOS telles que les « Collections » et « mappingRelation » qui permettent respectivement de regrouper les concepts provenant d'ensembles de données terminologiques différents et de faire la

⁵ GUI : « Graphical user interface »

correspondance entre eux. Par ailleurs, il n'offre pas non plus la possibilité de s'adapter aux évolutions de la structure SKOS comme par exemple l'extension SKOS-XL⁶.

2.4.3 SKOS Editor

« SKOS Editor » (Simon Jupp et al., 2008) est un autre outil qui permet la gestion des ensembles terminologiques formalisés sous SKOS et qui est destiné à être connecté à l'outil ontologique « Protégé »⁷ (Natalya Noy et al., 2000). « SKOS Editor » est le produit des efforts des chercheurs Simon Jupp et al. (2008). Ces derniers ont travaillé sur la réalisation d'une API et d'un plugin « SKOS Editor » permettant de gérer des vocabulaires contrôlés normalisés en SKOS.

Le plugin « SKOS Editor » englobe toutes les fonctionnalités de SKOS. Grâce à l'utilisation du module *SKOSManager*, « SKOS Editor » permet de charger et de créer différents ensembles terminologiques normalisés en SKOS. Il permet, par ailleurs, de sauvegarder des séries de données SKOS et de les visualiser sous une forme hiérarchique.

Cet outil offre plusieurs avantages, dont principalement la bonne gestion des différentes formes de systèmes de gestion des connaissances, qui a été d'ailleurs démontrée grâce à des tests sur de multiples ensembles terminologiques. Néanmoins, « SKOS Editor » a été élaboré pour être monté sur un outil qui gère principalement les ontologies qui sont une forme très complexe d'ensembles terminologiques. Par ailleurs, cet outil est difficile d'utilisation notamment concernant l'édition et la visualisation de modèles simples sous SKOS. De plus, lors de la création d'un nouvel ensemble terminologique, l'outil fournit différentes fonctionnalités qui, parfois, ne sont pas adaptées aux types de relations existantes dans SKOS (Javier Lacasta et al., 2007).

De plus, selon le rapport du « World Wide Web Consortium », SKOSed ne se conforme pas en totalité à la représentation théorique énoncée dans le document référence de SKOS. En effet, certaines propriétés ont été omises. Des propriétés, comme « *orderedCollection* », « *member* » et « *memberList* », n'ont pas été implémentées. Or, le document de référence les énonce clairement. L'omission de telles propriétés pourrait

⁶ www.w3.org/2008/05/skos-xl

⁷ <http://protege.stanford.edu/overview/>

restreindre son utilisation et altérer la qualité des données des ensembles de données terminologiques.

2.4.4 ONKI SKOS

L'ONKI SKOS Server (Jouni Tuominen et al., 2009) est, selon le « *National Finnish Ontology Service* », un serveur permettant de publier, d'intégrer et d'utiliser des méthodes d'accès rapide via le web à des vocabulaires SKOS ou à des ontologies. Il offre non seulement la possibilité de chercher et de visualiser des vocabulaires spécifiés en SKOS mais aussi les ontologies en RDFS.

Grâce à son interface graphique, le service ONKI permet une meilleure utilisation et une facilité de manipulation des concepts des thésaurus. Cette interface est constituée principalement de trois parties.

La première, « Semantic autocompletion concept search », est la partie dans laquelle, l'utilisateur saisit le concept recherché. Une fois que le nom du concept recherché est introduit, une requête est envoyée pour associer la chaîne de caractères saisie à un ou plusieurs concepts. Une fois le ou les concepts identifiés, le résultat est affiché dans la seconde partie. La deuxième partie, dite « Concept Hierarchy », contient la hiérarchie du concept sélectionné. La hiérarchie est générée en se basant sur les propriétés (*skos:narrower*, *skos:broader*, *skos:hasTopConcept*) définies dans le thésaurus par le langage SKOS. La troisième partie de l'interface graphique affiche les propriétés du concept qui lui sont associées, comme par exemple, la taille d'une ville, d'une population, etc.

Hormis le fait de permettre de chercher et de consulter des concepts, l'ONKI permet de sélectionner des références de concepts et de les enregistrer dans une base de données d'une application en utilisant HTML ou JavaScript.

Ces différentes opérations n'ont pu être rendues possibles que grâce à la publication de thésaurus sur le serveur ONKI, qui d'ailleurs n'était pas une tâche évidente, notamment parce que les thésaurus provenaient de sources diverses. Pour ce faire, il a fallu développer un processus de conversion permettant de transformer n'importe quel thésaurus en SKOS.

Le processus de transformation consiste essentiellement à exporter un thésaurus depuis un autre système vers un fichier XML en se basant sur un langage script tel que MARCXML, XML dumps, etc. Une fois le fichier XML réalisé, le convertisseur se charge d'effectuer les traitements nécessaires afin de produire un fichier SKOS. En effet pour parvenir à créer ce fichier, le convertisseur charge le fichier XML. Puis, il génère une structure et un domaine de noms nécessaires pour le modèle SKOS. Le convertisseur s'occupe ensuite de transformer les relations définies dans le fichier XML par leurs contreparties dans le langage SKOS. Finalement, un fichier contenant le groupement des concepts est joint au convertisseur permettant ainsi la création du fichier RDF qui sera envoyé vers le serveur ONKI, pour que les utilisateurs puissent enfin l'utiliser.

L'ONKI SKOS présente, certes, plusieurs avantages dont le plus important est le convertisseur. Néanmoins, il demeure que plusieurs inconvénients ont été recensés dont principalement sa difficulté d'utilisation. En effet, il est difficile à un utilisateur non averti de créer de manière intuitive et interactive son thésaurus. La création du thésaurus sous le format SKOS doit être réalisée au préalable par un expert.

2.5. CONCLUSION

Les ensembles terminologiques, malgré qu'ils soient bien identifiés, donnent du fil à tordre à leurs utilisateurs. Malgré l'adoption du standard SKOS pour leur normalisation, les outils qui reposent sur ce standard restent inadéquats et difficiles d'utilisation. La majorité de ces outils, selon le W3C, ne se conforment pas à la description théorique du standard SKOS. Un rapport émis par cet organisme le démontre et résume ces manques grâce à un tableau comparatif (<http://www.w3.org/2006/07/SWD/SKOS/reference/20090315/implementation.html>). Par ailleurs, après étude de ces outils, nous avons aussi constaté que bon nombre d'entre eux n'offraient pas l'extension de SKOS, SKOS-XL, alors que cette dernière permet de donner plus de flexibilité aux utilisateurs et surtout, plus de souplesse aux ensembles de données terminologiques qui dans bien des cas, ne se conforment pas lors de leur création au standard qui s'y rapporte.

Aussi, pour remédier à ces limites, nous proposons une approche basée sur la gestion de modèles pour concevoir et prototyper un outil plus adapté aux besoins des utilisateurs.

CHAPITRE III

OBJECTIFS ET MÉTHODOLOGIE DE RECHERCHE

3.1. INTRODUCTION

Dans le chapitre précédent, nous avons procédé à une revue de la littérature des trois principaux éléments qui touchent à la compréhension de notre domaine d'études. Nous avons, en outre, présenté les différents types d'ensembles de données terminologiques ainsi que les standards qui les régissent. Par ailleurs, nous avons exposé une liste d'outils permettant leur gestion et manipulation tout en citant leurs limites.

Afin de pallier aux insuffisances de ces outils, nous proposons une approche basée sur la gestion de modèles pour concevoir et développer un outil plus adapté aux besoins utilisateurs. Notre travail de recherche a donc pour objectifs :

1. L'identification des différents éléments qui permettent une meilleure interopérabilité des ensembles de données terminologiques.
2. La validation du modèle conceptuel de SKOS qui a été réalisé lors des travaux précédents.
3. La conception et la validation d'un outil logiciel facilitant la gestion d'ensembles de données terminologiques et leur échange entre différentes communautés. Cet outil permet de créer, visualiser et transférer des systèmes d'organisation des connaissances basés sur SKOS ainsi qu'à simplifier la recherche de données.

Dans ce chapitre, nous présentons la méthodologie de recherche employée et nous détaillons les différentes activités permettant d'atteindre les objectifs que nous nous sommes fixés.

3.2. PRESENTATION DE LA DSRM OU « DESIGN SCIENCE RESEARCH METHODOLOGY »

Les méthodes de recherche ont été souvent négligées dans les sciences de la conception et des systèmes d'information. Différents chercheurs, tels que Nunamaker et al. ou Walls et al. se sont penchés sur l'étude de méthodologie de recherche. Quelques uns ont proposé de développer une méthode permettant de guider les recherches en science de la conception et plusieurs d'entre eux ont présenté quelques orientations pour définir ces recherches et pour diffuser des références d'ouvrages.

Parmi ces chercheurs, Ken Peffers et al. (2007) ont proposé la DSRM ou « Design Science Research Methodology » comme méthodologie de recherche.

La DSRM vise à fournir un modèle de processus nominal cohérent et un modèle mental - une description du fonctionnement d'un système et le comportement de ses objets - permettant de présenter et d'évaluer les recherches dans le domaine des systèmes d'information. Elle tend aussi à aider les chercheurs à produire et à présenter une bonne qualité de recherches afin de valoriser leurs travaux.

Le modèle de processus de la DSRM repose sur six activités. Selon le type de recherche, les chercheurs ont le choix de commencer par l'une ou l'autre de ses quatre premières activités. Par exemple, si la recherche est centrée sur le problème, ils commenceront par la première activité, que nous développons dans la suite du document.

Par ailleurs, ce modèle permet, grâce au processus d'itération, de raffiner les solutions obtenues et/ou de corriger des erreurs décelées lors des phases d'évaluation et de communication.

Comme nous l'avons annoncé plus haut, la DSRM se compose de six activités que nous présentons dans la suite.

3.2.1 Activité 1 : Identification du problème.

Cette activité vise à définir le problème spécifique de la recherche et à justifier la valeur d'une solution. La définition du problème sera utilisée pour développer un artefact qui pourrait être considéré comme une solution alors que la justification de la valeur d'une solution permettra:

- De motiver les chercheurs et le public de la recherche à poursuivre la solution et à accepter les résultats.
- D'aider à comprendre le raisonnement associé à la pensée du chercheur vis-à-vis du problème.

3.2.2 Activité 2 : Définir les objectifs de la solution

Cette activité a été inspirée par les travaux de Nunamaker et al.,(1991) et Archer (1964). Lors de cette étape, il y a lieu de déduire les objectifs réalisables et possibles de la solution à partir de la définition du problème de la recherche qui a été réalisée lors de l'activité 1.

3.2.3 Activité 3 : Conception et développement

Cette activité a été inspirée par les travaux de Hevener et al. (2004) et de P. Jarvinen (2007). Au cours de l'étape de conception et développement, nous déterminons les fonctionnalités souhaitées de l'artefact et son architecture. Aussi, il y a lieu d'identifier les ressources requises pour passer des objectifs à la conception et au développement, comme par exemple, les différentes théories qui pourraient être mises à profit dans une solution.

3.2.4 Activité 4: Démonstration

Pendant l'étape de démonstration, les chercheurs démontrent la capacité de l'artefact à résoudre une ou plusieurs instances du problème et ce, en employant des méthodes telles que :

- Une ou plusieurs expérimentations.
- Simulation.
- Études de cas.

3.2.5 Activité 5: Évaluation

La phase d'évaluation permet d'observer et de mesurer à quel point l'artefact soutient une solution à ce problème. Cette activité consiste à comparer les objectifs d'une solution et à observer les résultats réels de son utilisation. La comparaison des fonctionnalités de l'artefact avec les objectifs de la solution est réalisée à l'aide de mesures qualifiables de performance tels que :

- Les résultats des enquêtes de satisfaction
- La rétroaction des clients

Et de mesures quantifiables des performances du système tels que:

- Les temps de réponse
- La disponibilité

3.2.6 Activité 6 : Communication « Archer, Hevner et al »

Peppers et al. (2007) se sont inspirés des travaux réalisés par Archer et al. (1984) et Hevner et al. (2004) pour l'élaboration de cette activité. Lors de cette dernière, les chercheurs communiquent sur le problème et son importance, l'artefact, son utilité et sa nouveauté, la rigueur de sa conception et son efficacité, pour les chercheurs et autres publics concernés.

3.2.7 Conclusion

Les activités citées ci-dessous forment la méthodologie de recherche DSRM. Celle-ci sera utilisée pour la réalisation de notre travail de recherche. La figure qui suit illustre l'enchaînement de ces différentes activités.

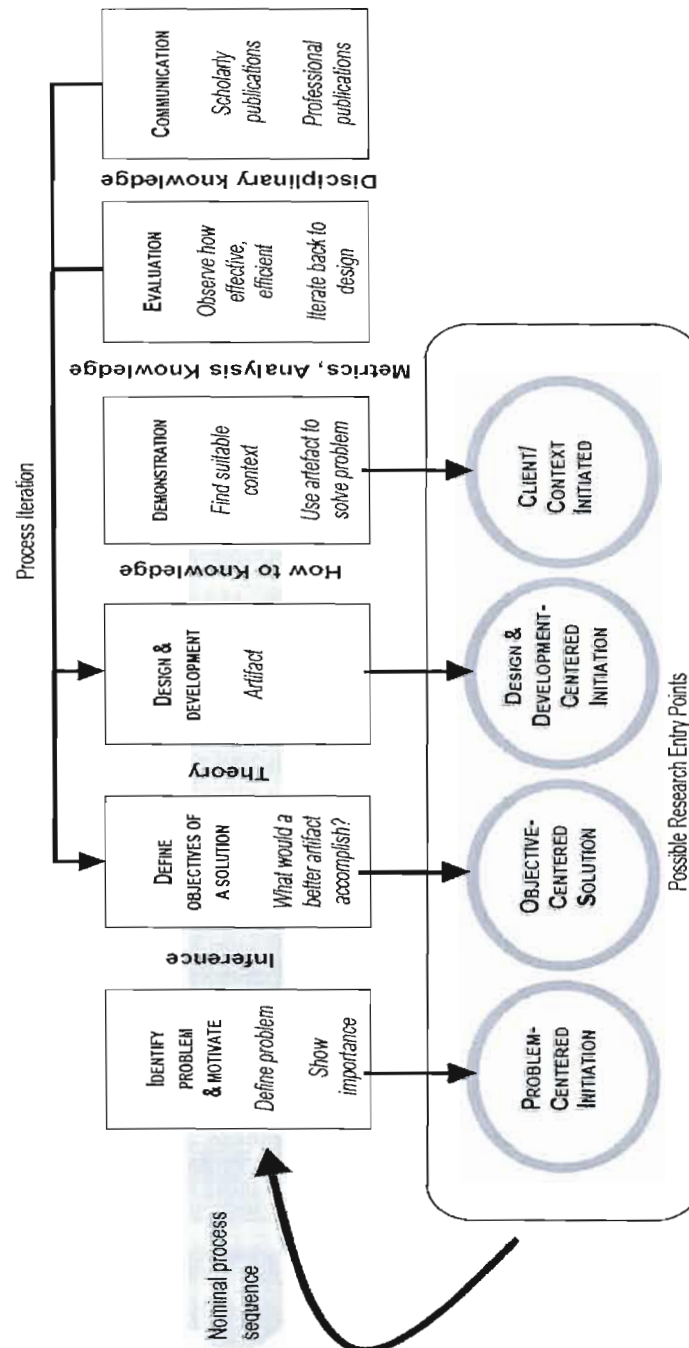


Figure 3.1 - Modèle de processus du « Design Science Research Method » (Ken Peffers et al., 2007)

3.3. PROJET DE RECHERCHE

Le projet de recherche que nous voulons réaliser consiste en la conception et le développement d'un outil apte à gérer des ensembles de données terminologiques basés sur SKOS. Le fait de développer un tel outil n'est pas une tâche facile. La difficulté principale, comme nous l'avons souligné auparavant, consiste principalement dans la mise en correspondance entre les propriétés de SKOS et celle des modèles conceptuels des systèmes de gestion des connaissances tels que les thésaurus.

Afin de franchir cet obstacle et afin d'aboutir à la production de notre outil, nous suivons donc le processus proposé par la DSRM. Celui-ci, en l'appliquant à notre projet de recherche permet d'aboutir aux activités suivantes.

3.3.1 Activité 1: Identification du problème

La première activité est sans doute l'activité qui a été des plus importantes. En effet, lors de cette activité, nous sommes arrivés à déterminer et à expliciter le problème que notre étude tente de résoudre. Lors de cette activité, nous avons aussi pu mettre un contexte à notre problème et déterminer notre domaine d'étude. Le travail réalisé est synthétisé dans les premier et deuxième chapitres de ce mémoire.

3.3.2 Activité 2: Définition des objectifs de la solution

Comme présenté dans la section 3.2.2, au cours de cette activité, il y a lieu de définir les objectifs visés par l'élaboration de notre outil. Ainsi pour cette activité, nous avons donc présenté les objectifs à atteindre d'une manière implicite et explicite respectivement lors des premier et troisième chapitres. Ces objectifs, rappelons-le, sont au nombre de trois et sont repris dans ce qui suit :

1. identifier les éléments qui permettent une meilleure interopérabilité des ensembles terminologiques ;
2. valider le modèle conceptuel de SKOS réalisé lors des travaux antérieurs ;
3. concevoir et valider un outil logiciel facilitant la gestion et l'échange de données terminologiques entre les différentes communautés ainsi que la

manipulation des systèmes d'organisation des connaissances basés sur SKOS.

3.3.3 Activité 3: Conception et développement

Le modèle conceptuel de SKOS réalisé lors d'un travail de recherche précédent constitue le point de départ de cette activité. En effet, avant de concevoir et de développer notre outil, nous serons amenés à le valider, tout d'abord, pour ensuite identifier les éléments qui pourraient correspondre aux propriétés des modèles conceptuels des systèmes de gestion des connaissances et ainsi avoir une base solide pour la conception et la réalisation de l'outil.

Une fois le modèle conceptuel validé et les différents éléments identifiés, nous commençons par définir l'architecture globale du système et ses fonctionnalités. Nous déploierons la base de données correspondante au modèle conceptuel de SKOS et aux besoins métiers de l'outil dans une étape plus avancée pour enfin procéder au développement de l'outil.

3.3.4 Activité 4 - 5: Démonstration & Évaluation

Dans notre travail de recherche, nous avons fusionné les deux activités de démonstration et évaluation en une seule activité afin d'expérimenter notre outil. Cette expérimentation a pour objectif de démontrer et d'évaluer la capacité de l'outil que nous avons développé lors de l'activité 3. Pour ce faire, nous introduisons des exemples d'ensemble de données terminologiques recueillis auprès d'organismes.

3.3.5 Activité 6: Communication

Cette activité consiste en l'étape finale de notre projet de recherche qui annonce la fin de notre travail d'étude où il sera question de communiquer nos résultats de recherche ainsi que nos conclusions à notre département et aux scientifiques intéressées au travers de notre mémoire de recherche et peut être d'un article scientifique.

3.4. CONCLUSION

La méthode que nous venons de décrire est celle qui servira à nous guider tout au long de notre recherche. En effet, les différentes activités que nous venons de présenter seront

utilisées pour réaliser l'ensemble de notre projet de recherche. Les activités une et deux ont été déjà réalisées dans les chapitres précédents.

Dans la suite, nous entamons l'activité trois de notre méthodologie et nous commençons par conceptualiser et développer notre outil, qui est l'artefact principal de cette étude. Cependant afin de produire un outil logiciel de qualité, nous débutons par valider le modèle conceptuel de SKOS réalisé lors des travaux précédents afin d'avoir une base solide pour l'accomplissement du reste du travail de recherche.

- - - - -

CHAPITRE IV

MÉTA-MODÈLE DE SKOS

4.1. INTRODUCTION

Comme nous l'avons vu précédemment, un de nos principaux objectifs consiste à concevoir et à prototyper un outil de gestion d'ensembles de données terminologiques basé sur SKOS. Pour ce faire, nous devons avoir une connaissance et une compréhension approfondies de ce standard afin de pouvoir, d'une part, le modéliser conceptuellement et d'autre part, produire un schéma de base de données le supportant et enfin développer un engin SKOS « SKOS engine » pour le stockage et la manipulation d'ensembles de données terminologiques.

Lors de ce chapitre, nous détaillons tout d'abord le standard SKOS, puis nous présentons le modèle conceptuel le supportant ainsi que les différentes propriétés qui le composent pour que nous puissions comprendre son modèle conceptuel. Ensuite, nous décrivons son schéma de base de données. Finalement, nous nous assurons de la validité du modèle conceptuel produit au moyen d'exemples retenus dans le document référence de SKOS rédigé par le W3C.

4.2. LE STANDARD SKOS

4.2.1 Historique

Le standard SKOS a été produit lors du projet européen « SWAD-Europe » (Dan Brickley et al., 2004) dans le cadre du programme Technologies de la Société de l'Information. Il avait pour objectif de soutenir l'activité Web sémantique du W3C.

A la fin de ce projet, le travail sur SKOS a été poursuivi par un groupe de travail du W3C lors d'une étude sur les bonnes pratiques et le déploiement des standards RDF (Semantic Web Deployment Working Group, 2006). Cette étude qui a duré environ quatre années a permis de renforcer le noyau de SKOS et a abouti en 2009 à une recommandation du W3C. Tout au long de cette période, différentes versions du document de référence sont apparues dont trois qui étaient candidates pour la recommandation.

Cette multiplicité des versions du document de référence affecte tout travail de recherche se basant sur SKOS. En effet, étant donné que ce document est sujet à des modifications, tout travail réalisé, d'une version à une autre du document de référence, peut être sujet lui aussi à des changements. Ainsi, dans le cadre de notre étude, nous avons dû revoir et apporter des modifications au travail de recherche précédent. Pour ce faire, nous nous sommes replongés dans la lecture du document du W3C et dans la compréhension des recommandations émises qui souvent, étaient ambiguës.

4.2.2 Les éléments de SKOS

Comme nous l'avons précisé lors du chapitre deux de ce mémoire, le standard SKOS repose sur la notion de « concept ».

Un « *Concept* » est une notion abstraite utilisée dans le standard SKOS pour exprimer des idées, des objets ou des événements (Summer et al, 2009). Il possède un « *Label* » et un « *URI* » (Uniform Resource Identifier) lui permettant d'être identifié d'une manière unique et simple (A.Miles et al, 2009). Ces « *Concepts* » sont la plupart du temps hiérarchisés au sein d'un « *Concept Scheme* » permettant de représenter un ensemble terminologique. Les « *Concepts* » peuvent aussi être regroupés au sein d'une ou plusieurs collections « *skos:Collection* » provenant ou non de divers « *Concept Scheme* ». Par ailleurs, il est possible de les ordonner grâce à l'utilisation des « *OrderedCollection* ».

4.2.2.1 Le concept SKOS

Comme nous l'avons souligné précédemment, le concept est l'élément central du standard SKOS. C'est une notion abstraite permettant de représenter la structure conceptuelle et intellectuelle des systèmes d'organisation des connaissances afin de décrire et transmettre le sens exact qui est établi à l'intérieur de ces systèmes.

Le standard SKOS représente ces concepts par l'utilisation de la classe « *skos:concept* ». Ainsi pour illustrer que « MonConcept » est un concept SKOS, on procède comme suit :

< MonConcept > rdf:type skos:Concept

4.2.2.2 Les Labels SKOS

Le standard SKOS fournit trois propriétés principales pour référer les concepts. Le « *skos:prefLabel* » ; « *skos:altLabel* » ; « *skos:hiddenLabel* ». Chacune de ses propriétés donne un statut spécifique pour le label d'un concept.

Skos:prefLabel : permet d'assigner à un concept un label préféré ; c'est le label qui décrit au mieux le concept auquel il est rattaché. Dans les thésaurus, ce label correspond généralement à un terme descripteur utilisé pour l'indexation. Il est à noter qu'il n'existe qu'un seul terme associé au « *skos:prefLabel* » par langue.

Exemple :

biologie rdf:type skos:Concept;

skos:prefLabel "biologie"@fr;

skos:prefLabel "biology"@en;

Skos:altLabel : permet d'assigner à un concept des synonymes, des abréviations ou des acronymes aux termes descripteurs, c'est-à-dire, au « *skos:prefLabel* » afin d'affiner la description d'un concept SKOS.

Exemple:

biologie rdf:type skos:Concept;

skos:prefLabel "biologie"@fr;

skos:altLabel "biote"@fr; (biote ensemble d'êtres naturels)

skos:prefLabel "biology"@en;

skos:altLabel "biota"@en;

Skos: hiddenLabel : est utilisé pour prévenir les fautes d'orthographe lors d'une saisie de termes de recherche et ainsi permettre de réaliser la recherche.

Exemple :

```
biologie rdf:type skos:Concept;

skos:prefLabel "biologie"@fr;

skos:altLabel "biote"@fr;

skos:hiddenLabel "byote"@fr;
```

4.2.2.3 Les schémas de concepts

Les schémas de concepts contiennent un ou plusieurs concepts, d'un domaine particulier. Ces derniers y sont hiérarchisés ainsi que les différentes relations qui peuvent subsister entre eux. Un schéma de concepts représente, généralement, un ensemble de données terminologiques tels que les thésaurus, les schémas de classification, etc.

Dans le standard SKOS, les schémas de concepts sont représentés à travers l'utilisation de la classe « *skos:ConceptScheme* ». Cette classe regroupe aussi trois principales propriétés :

skos: inScheme : c'est une propriété qui permet de spécifier qu'un concept « *skos:Concept* » appartient à un schéma de concepts « *skos: ConceptScheme* ».

skos:hasTopConcept : Il sert à lier un schéma de concepts « *skos:ConceptScheme* » avec un concept « *skos:Concept* » afin de le définir comme étant le plus haut de la hiérarchisation au sein du schéma de concepts.

skos:topConceptOf : à l'inverse de « *skos:hasTopConcept* », cette propriété permet de déterminer qu'un concept donné est classifié au plus haut niveau de la hiérarchie d'un schéma de concepts.

Exemple :

```
biologie rdf:type skos:Concept;
```

anatomie **rdf:type skos:Concept;**

< MonScheme > **skos :hasTopConcept** *biologie*

<*anatomie*> **skos :inScheme** <MonScheme >

4.2.2.4 Les Collections de concepts

Les collections de concepts sont utilisées pour regrouper plusieurs concepts, qui partagent des caractéristiques communes dans un même ensemble, sous un même label ou pour les classer selon un ordre significatif.

Pour pouvoir utiliser des collections de concepts, on recourt généralement au « *skos :Collection* » ou au « *skos :orderedCollection* ». Ce dernier est sous forme de collection contenant des concepts disposés selon un ordre spécifique. Afin de grouper ou d'ordonner les concepts à l'intérieur de ces collections, on utilise respectivement les propriétés « *skos :member* » et « *skos :memberList* ».

Exemple 1:

<génomique> **rdf: type skos :Collection ; skos :member** <génom>,
<gène>, <génotype>

Exemple 2:

<génomique> **rdf: type skos : OrderedCollection ; skos :memberList**(<gène>
<génom> <génotype>)

4.2.2.5 La documentation

Dans le standard SKOS, la documentation est réalisée à l'aide de notes « *skos :note* » fournissant les informations relatives aux concepts et permettant, par exemple, d'attribuer une définition, de présenter un exemple, ou plus encore d'assigner une image aux concepts.

La classe « *skos :note* » compte différentes propriétés :

Skos:changeNote procure de fines informations sur les changements effectués sur les concepts.

Exemple :

<tomate> skos:changeNote "Move from under 'fruits' to under 'vegetables' by Thomas"@en.

Skos:definition attribue une explication complète du sens donné au concept, ou au terme que le concept représente.

Exemple :

<documentation> skos:definition "the written comments, graphical illustrations, flowcharts, manuals, etc., supplied with a program or software system"@en.

Skos:editorialNote sert à aider les personnes en charge d'effectuer des mises à jour en fournissant des notes permettant par exemple de les avertir qu'il reste du travail éditorial à compléter sur le concept.

Exemple :

<Biota> skos:editorialNote "Check spelling with Thomas"@en

Skos:example donne un exemple d'utilisation du concept auquel il est rattaché.

Exemple :

<organizationsOfScienceAndCulture> skos:example "academies of science, general museums, world fairs"@en.

Skos:historyNote donne une description de l'ensemble des changements qui ont été effectués sur un concept donné.

Exemple :

<childAbuse> skos:historyNote "estab. 1975; heading was: Cruelty to children [1952-1975]"@en

Skos:scopeNote fournit des informations sur le sens voulu d'un concept, ainsi que la façon dont l'utilisation de ce concept est limitée dans la pratique d'indexation.

Exemple :

<microwaveFrequencies> skos:scopeNote "Used for frequencies between 1GHz to 300Ghz"@en

4.2.2.6 Les relations sémantiques dans SKOS

Les relations sémantiques permettent de relier les concepts SKOS entre eux au sein d'un même ou différents schémas de concepts. Il existe plusieurs sortes de relations sémantiques en SKOS.

Premièrement, « *skos:related* » est une relation qui permet d'associer un concept à un autre. Il représente les relations associatives qui relient les différents concepts d'un même niveau, c'est-à-dire, à mettre en relation des termes qui sont semblables grâce par exemple à des relations synonymiques.

Deuxièmement, « *skos:broader* » et « *skos:narrower* » sont deux autres variétés de relations sémantiques. Celles-ci relient respectivement différents concepts entre eux grâce à des relations de généralisation, spécialisation. Il est à remarquer que ces deux éléments sont respectivement des spécialisations de « *skos:broaderTransitive* » et « *skos:narrowerTransitive* ».

Finalement, « *skos:mappingRelation* » permet de relier des concepts de différents schémas de concepts. Elle regroupe d'autres propriétés qui sont les suivantes :

- **skos:broadMatch** et **skos:narrowMatch** sont respectivement une spécialisation de « *skos:broader* » et « *skos:narrower* ». Ces propriétés permettent de relier deux concepts appartenant chacun à un schéma de concepts différent grâce à des relations hiérarchiques.

- **skos:relatedMatch** est une spécialisation de « *skos:related* » reliant deux concepts appartenant à deux schéma de concepts différents par une relation associative.
- **skos:closeMatch** est utilisé pour lier deux concepts qui sont assez similaires pour qu'il puissent être interchangeables lors de l'extraction de données.
- **skos:exactMatch** est employé pour lier deux concepts qui sont fortement semblables pour être utilisés lors de l'opération de l'extraction de données.

4.2.2.7 SKOS eXtension Label

SKOS eXtension Label ou *SKOS-XL* est une extension de SKOS. Elle fournit un support additionnel pour l'identification, la description et la mise en relation d'entités lexicales (A. Miles, 2009). La classe principale de cette extension est la « *skosxl:label* ». Elle est constituée de trois propriétés « *skosxl:prefLabel* », « *skosxl:altLabel* » et « *skosxl:hiddenLabel* » qui sont analogues à celles de « *skos:label* ». Par contre, cette classe offre la possibilité de relier directement des « *skosxl:label* », grâce à la propriété « *skosxl:labelRelation* » pour définir de nouveaux types de relations non supportées par SKOS comme par exemple les acronymes. La figure suivante présente une description de cet élément de SKOS.

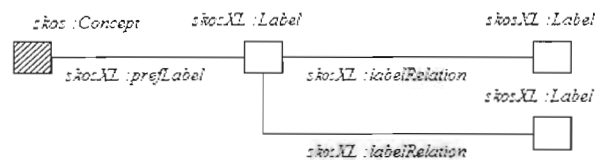


Figure 4.1 – Description du fonctionnement des propriétés de SKOSXL

4.2.3 Évolution du standard

Grâce au document référence de SKOS, nous sommes parvenus, d'une part, à identifier les différentes propriétés de SKOS et à les présenter dans cette section. Les définitions

données par le W3C sont dans l'ensemble claires. Cependant, il en demeure que certaines restent ambiguës et cela pose problème pour les conceptualiser d'une manière précise.

D'autre part, le standard SKOS est sujet à évolution ou/et modification et devrait donc être ajusté au fur et à mesure que de nouvelles évolutions apparaissent.

4.3. MODELISATION CONCEPTUELLE DU STANDARD

Le modèle conceptuel de SKOS est une représentation formelle du standard SKOS qui a été réalisée à partir du document de référence de SKOS fourni par le W3C. Il reproduit toutes les différentes propriétés énoncées dans le standard ainsi que les règles qui s'y appliquent. Ce modèle a été schématisé en utilisant une représentation UML. Ce modèle se compose de deux parties. Une première qui correspond aux entités du modèle conceptuel de SKOS et une seconde qui définit les relations existantes entre ces entités.

La figure ci-dessous représente la première partie de ce modèle conceptuel.

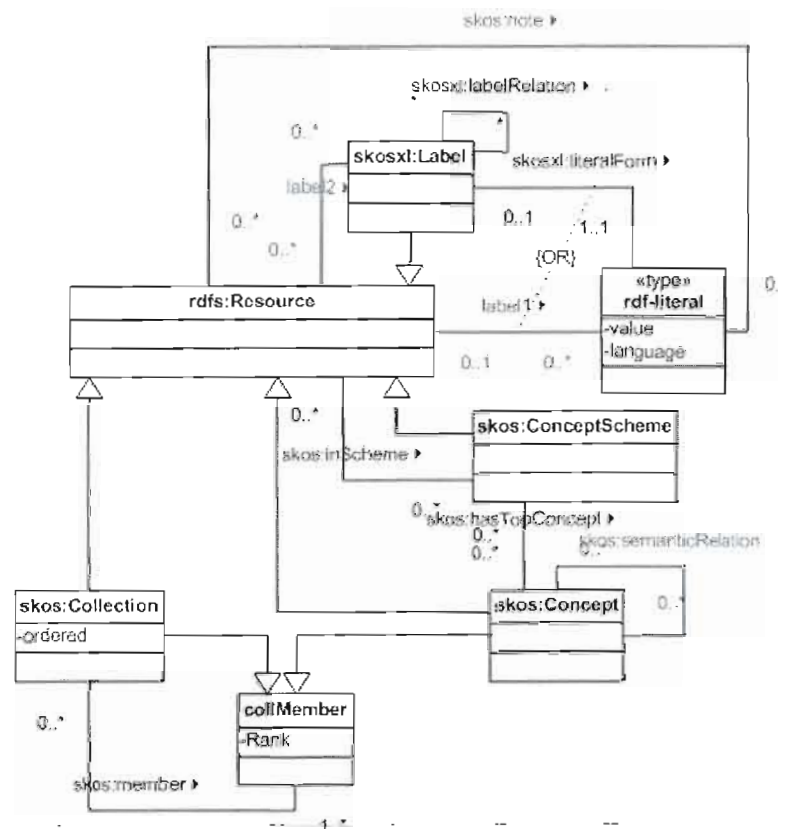


Figure 4.2 – Les entités du modèle conceptuel de SKOS

A partir de cette figure, nous pouvons distinguer les principaux éléments qui constituent le standard SKOS telles que les « Concepts », les « ConceptScheme », les « Collections » et les « OrderedCollections » et les « SKOSXL : Label » qui sont des « *rdfs : Resource* », tel qu'il a été énoncé dans le document de référence de SKOS.

Le modèle conceptualisé contient les différentes classes du méta-modèle SKOS présenté auparavant dont les principales sont « *skos:concept* », « *skos:conceptScheme* » et « *skos:collection* » ainsi que l’extension du label SKOS représentée ici par « *skosxl:Label* ». Celles-ci sont, comme nous l’avons cité lors de la présentation du méta-

modèle, des ressources RDFS⁸. Elles héritent, par conséquent des propriétés de l'entité « *rdfs:resource* » grâce aux relations de généralisation-spécialisation.

Outre ces classes, nous nous sommes intéressés aux éléments « *skos:collection* » et « *skos:orderedCollection* ». En effet, la différence entre les deux est que l'une est ordonnée alors que l'autre ne l'est pas. Donc pour distinguer les deux et pour raffiner notre modèle, nous avons rajouté l'attribut « *ordered* » qui prendrait une valeur booléenne en fonction de la nature de la collection. Aussi, avons-nous ajouté l'association « *collMember* » qui permet non seulement à un concept d'être membre d'une collection mais aussi à une autre collection de l'être. En occurrence, c'est pour cette raison que nous avons spécialisé les entités « *skos:collection* » et « *skos:concept* » à partir de « *collMember* ».

Par ailleurs, étant donné qu'une collection ordonnée doit contenir des concepts ou/et des collections, nous avons utilisé d'une part l'attribut « *rank* » pour leur attribuer un numéro d'ordre et d'autre part, l'association « *skos:member* » pour les lier et ainsi respecter la règle S32 du document de référence qui stipule « the *rdfs:range* of *skos:member* is the union of classes *skos:Concept* and *skos:Collection* ». Ainsi, nous sommes arrivés à la transformation suivante :

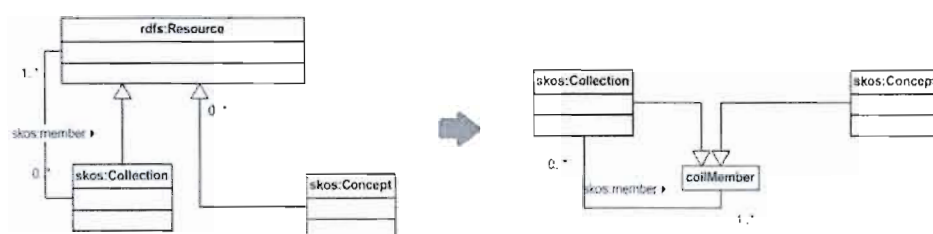


Figure 4.3 - Transformation de l'association « *skos:member* »

Le modèle conceptuel de SKOS contient en plus des entités, plusieurs relations représentant pareillement des éléments de SKOS.

⁸ RDFS ou RDF Schema appartient à la famille des langages du web sémantique du W3C. Il est destiné à structurer des ressources RDF.

Afin d'établir les différentes associations, nous avons débuté par la contrainte de base du méta-modèle qui spécifie qu'un concept peut être libre ou faire partie soit d'une collection soit d'un schéma de concepts. Pour ce faire, nous avons réussi tout d'abord, grâce aux éléments « *skos:member* » pour les collections (voir plus haut) et « *skos:inScheme* » pour les schémas de concepts, à représenter les liens d'appartenance pour un concept. Ensuite, en recourant au « *skos:hasTopConcept* », le modèle conceptuel de SKOS offre à un concept d'un schéma de concepts d'être son élément de haut niveau. L'association de même nom liant l'entité « *skos:concept* » et « *skos:conceptScheme* » représente cet élément.

Par ailleurs, à travers les associations « *label1* » et « *label2* », nous pouvons non seulement associer à un concept SKOS les différents libellés autorisés par SKOS mais aussi celles de son extension SKOSXL. Les deux associations « *label1* » et « *label2* » permettent respectivement de représenter les propriétés :

- « *skos:altLabel* », « *skos:prefLabel* » et « *skos:hiddenLabel* »
- « *skosxl:prefLabel* », « *skosxl:altLabel* » et « *skosxl:hiddenLabel* »

Certes, nous pouvons constater que les propriétés de chacune des associations « *label1* » et « *label2* » sont semblables. Cependant, en nous référant au document de référence, celui-ci spécifie que chacune des propriétés représentées dans l'association « *label2* » devra être liée à un « *skosxl:Label* » afin de permettre la création de nouvelles relations « *skosxl:labelRelation* » entre les différents « *skosxl:Label* ». Par ailleurs, ce même document spécifie que pour permettre l'association d'une chaîne de caractères valorisant un « *skosxl:Label* », il faut recourir à la relation « *skosxl:literalForm* ». Cette dernière relie un « *skosxl:literalForm* » à un « *rdf-literal* » contenant la valeur du « *skosxl:Label* » ainsi que la langue employée pour décrire cette valeur.

De plus, grâce à l'association « *skos:note* », notre modèle permet d'affecter des notes à n'importe quelle ressource SKOS. Cette association permet par exemple d'associer des exemples ou des définitions ou des notes d'édition pour documenter une ressource SKOS. Elle regroupe aussi plusieurs autres types de propriétés, qui sont visibles dans la figure 4.4.

L'utilisation de l'association « *skos :semanticRelation* » permet au modèle de supporter la propriété SKOS de même nom. Celle-ci rappelle qu'elle permet de lier des concepts grâce à des relations telles que hiérarchiques, etc. Cette association regroupe des sous-associations permettant de satisfaire les diverses propriétés suggérées par « *skos :semanticRelation* ». Celles-ci sont aussi visibles dans la figure 4.4, ci-dessous. Cette figure contient les diverses sortes d'associations que nous avons utilisées pour la réalisation du modèle conceptuel et est représentée dans ce qui suit.

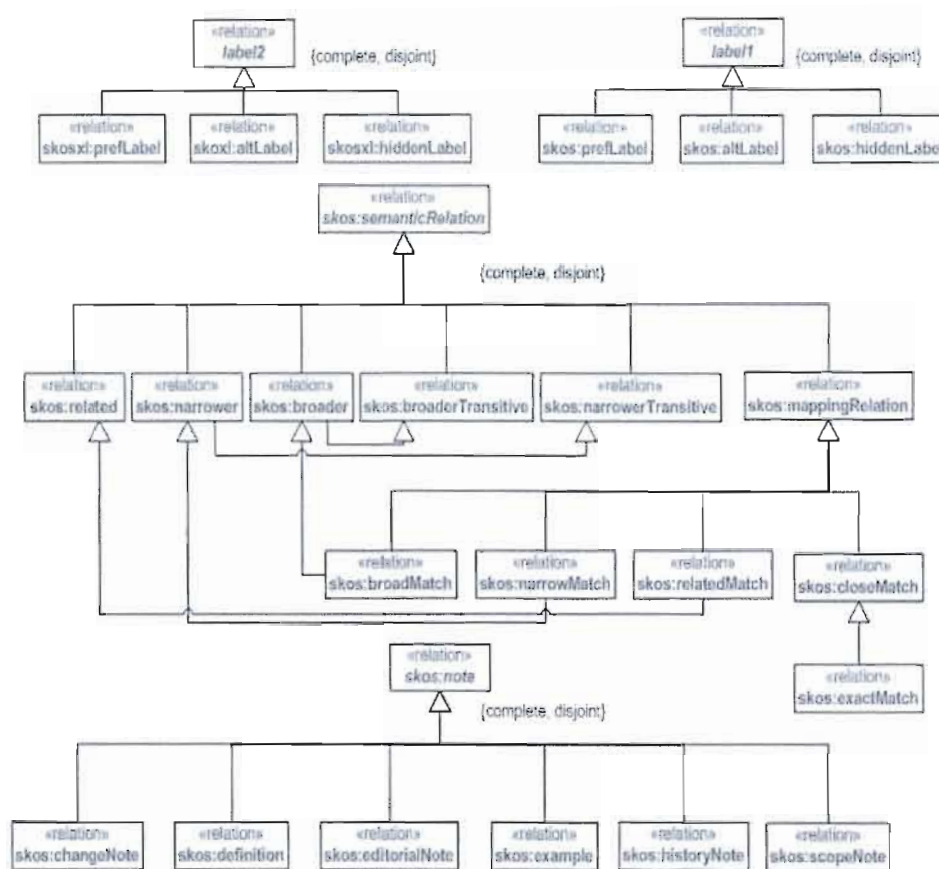


Figure 4.4 - Les relations du modèle conceptuel

4.3.2 Conclusion

Le modèle conceptuel que nous venons de présenter reflète le standard SKOS. Toutes les propriétés qu'il regroupe, ont été reprises dans notre modèle conceptuel en recourant à

des entités et à des associations. Ce modèle par ailleurs a été optimisé en tenant compte des contraintes spécifiées par SKOS. L'une des optimisations concerne notamment les associations liant les entités. D'ailleurs, c'est pour cette raison que nous avons découpé le modèle conceptuel en deux parties. La figure qui suit représente les deux parties que nous venons de présenter et résume le modèle conceptuel de SKOS qui a été réalisé.

-- -- -- -- -- -- -- -- -- --

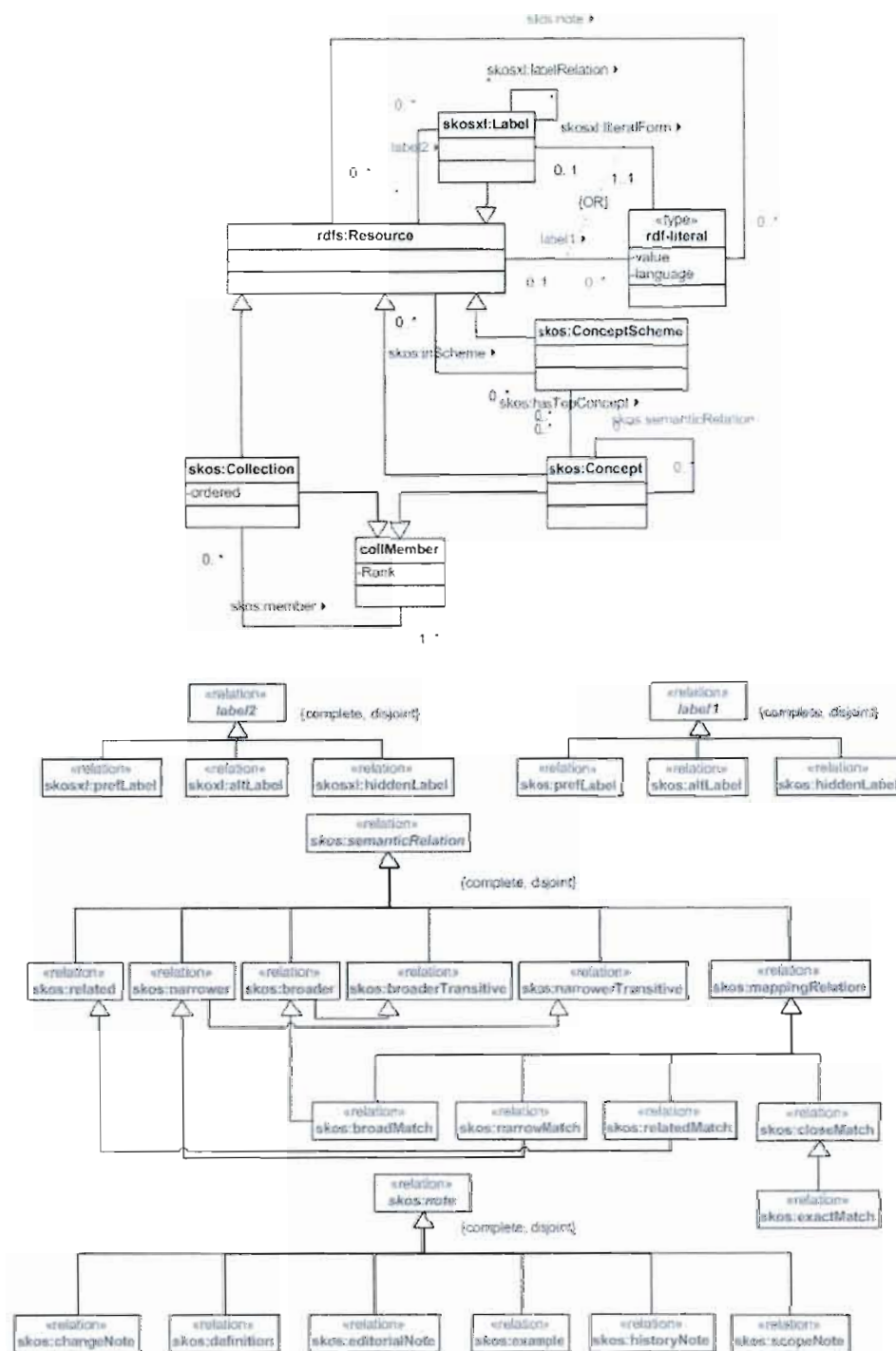


Figure 4.5 - Modèle conceptuel de SKOS et ses relations

4.4. SCHEMA DE LA BD SUPPORTANT SKOS

Une fois le modèle conceptuel finalisé, nous avons procédé à sa transformation en un schéma de base de données relationnelle. Nous avons repris les mêmes entités et transformé certaines des associations existantes en des entités. Certaines entités telles que « Langue » ou « noteType » ou « semanticRelationType » ont été rajoutées pour convenir à des spécifications métiers. Celles-ci contiennent les propriétés de chaque association que nous avons présentée dans la figure 4.4. Le dessin suivant donne une idée globale sur le schéma de la base de données SKOS que nous allons reproduire lors de l'activité de « conception et développement » de la méthodologie que nous appliquons.

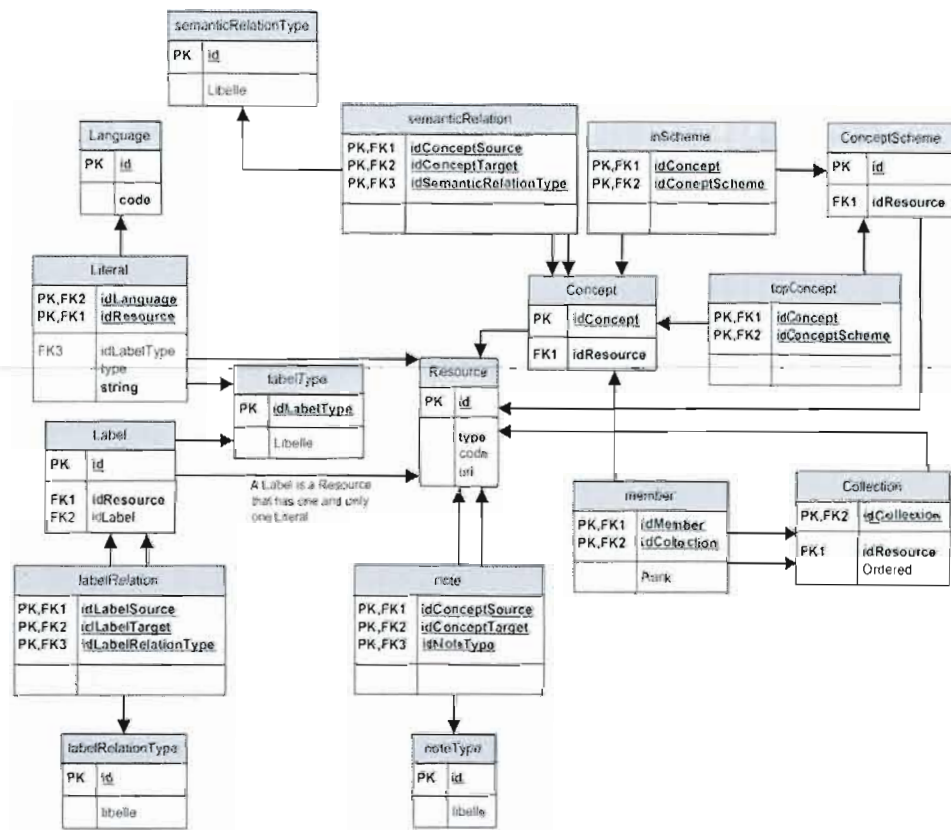


Figure 4.6 - Schéma de la base de données SKOS

Ce schéma de base de données est représentatif du modèle conceptuel de SKOS produit précédemment. Néanmoins, nous ne nous sommes pas assurés que le modèle sur lequel il repose est valide. Certes, le modèle conceptuel de SKOS semble supporter l'ensemble des

contraintes imposées par le méta-modèle étant donné qu'il prend en considération les contraintes énoncées dans le document référence de SKOS. Néanmoins, il est impératif que nous validions ce modèle avant que nous passions à l'étape suivante de la DSRM qui consiste, rappelons-le, à concevoir et à prototyper l'outil.

4.5. VALIDATION DU MODELE CONCEPTUEL DE SKOS

Après avoir présenté le modèle conceptuel de SKOS ainsi que le schéma de la base de données, nous sommes passés à l'étape de validation de ce modèle. Pour ce faire, nous avons pris plusieurs exemples qui ont été utilisés par le W3C pour documenter le standard SKOS ainsi que des exemples procurés par l'observatoire international des coopératives de services financiers⁹, et nous les avons appliqués à notre modèle conceptuel. Les exemples ont été choisis d'une façon à pouvoir valider les différentes propriétés modélisées dans notre modèle conceptuel et ont servi à la réalisation de diagrammes d'objets.

Il est à rappeler que le diagramme d'objet, dans le langage de modélisation de données UML, nous permet de modéliser des données et des relations existantes entre des instances de classes afin d'exprimer des contextes d'exécution (Laurent Audibert, 2009). Dans notre étude, ils serviront aussi à vérifier et à valider notre méta-modèle SKOS.

Les diagrammes d'objets ont été réalisés pour chaque grande catégorie de propriétés citée dans le document référence.

4.5.1 Schéma de concepts :

Afin d'examiner la conformité de la propriété « conceptScheme » avec notre modèle conceptuel de SKOS, nous avons choisi l'exemple 5 du document de référence représenté dans la figure ci-dessous et nous l'avons appliqué sur ce dernier.

⁹ <http://observatoire.coopfinance.hec.ca/>

Example 5 (consistent)

```

<MyScheme> rdf:type skos:ConceptScheme ;
  skos:hasTopConcept <MyConcept> .

<MyConcept> skos:topConceptOf <MyScheme> .

<AnotherConcept> skos:inScheme <MyScheme> .

```

Figure 4.7 - Exemple consistant d'un concept schème

La représentation de cet exemple à travers notre modèle n'a pas posé de problème. Le diagramme ci-dessous démontre que notre modèle le supporte parfaitement.

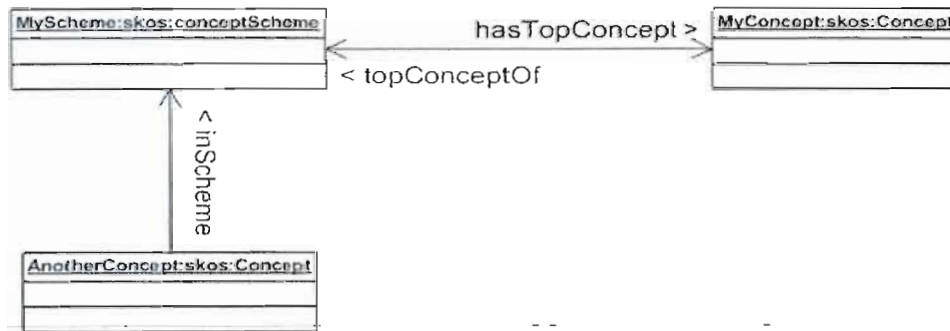


Figure 4.8 - Diagramme d'objet de l'exemple 5

4.5.2 Labels lexicaux

Afin de valider la propriété « Lexicallabel », nous avons utilisé l'exemple 10 reproduit dans la figure 4.9. L'exemple décrit une ressource ayant pour libellés préférés « animals » en anglais et « animaux » en français et possédant comme libellés alternatifs « fauna » en anglais et « faune » en français. Cette ressource possède aussi un libellé caché « aministrals » dans le cas où lors de la saisie l'orthographe du mot a été mal écrite.

Exemple 10 (consistant)

```

<MyResource>
  skos:prefLabel "animals"@en ;
  skos:altLabel "fauna"@en ;
  skos:hiddenLabel "aminals"@en ;
  skos:prefLabel "animaux"@fr ;
  skos:altLabel "faune"@fr .

```

Figure 4.9 - Exemple consistant d'un concept labélisé

L'instanciation et la représentation de cette ressource, à travers le diagramme dans la figure qui suit, ont été aisément illustrées à travers l'association « label1 » désignant le type du label et la classe « Literal » permettant de leur associer des valeurs. Nous remarquerons que les différentes propriétés, qui y ont été employées, ont leur équivalent dans le modèle produit de SKOS.

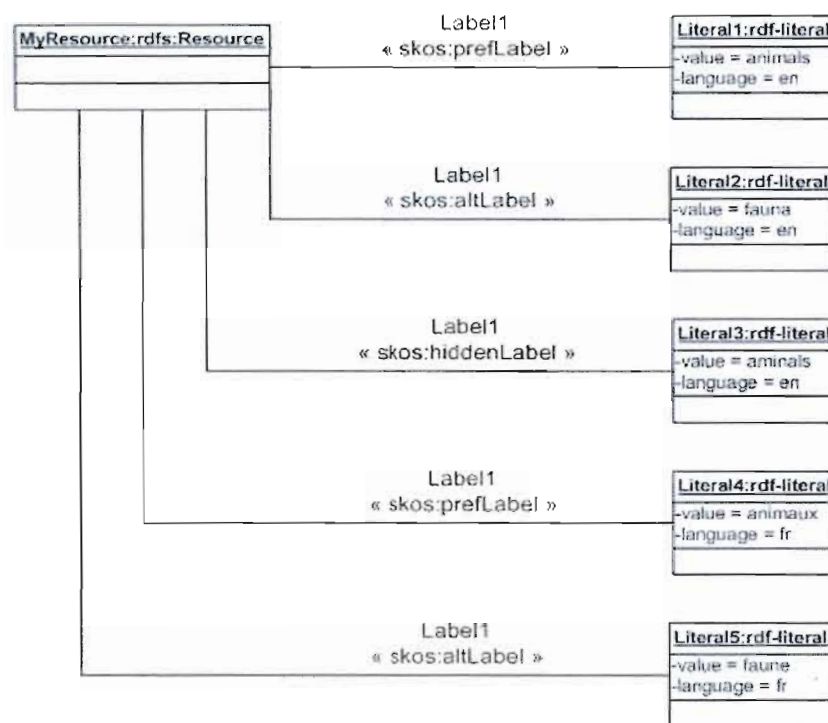


Figure 4.10 - Diagramme d'objet de l'exemple 10

4.5.3 Documentation

Concernant les notes relatives aux ressources « `rdf:resource` » nous traitons la validité de sa représentation en recourant à l'exemple ci-dessous.

Example 22 (consistent)

```
<MyResource> skos:note "this is a note"@en .
```

Figure 4.11 - Exemple consistant de la propriété "Note" de SKOS

Cet exemple nous a permis, dans un premier temps, de constater que notre modèle conceptuel contenait une erreur et nous a permis, dans un second temps, de la rectifier. L'association « `skos:note` » grâce auquel des notes entre des ressources « `rdfs:resources` » peuvent être représentées, ne permettait pas, en effet de stocker des valeurs à ces notes en utilisant le « `Literal` ». Dès lors, et comme alternative, nous avons dû changer le type de la relation existante entre « `rdfs:resource` » et « `rdf-literal` » en une relation de généralisation. Cette alternative nous permet d'attribuer une valeur à une note ainsi que la langue dans laquelle cette valeur a été donnée. Le diagramme d'objet correspondant est représenté dans la figure suivante.



Figure 4.12 - Diagramme d'objet correspondant au « `skos:note` »

4.5.4 Collection de concepts

Afin de valider notre modèle conceptuel concernant les collections de concepts, nous recourons aux exemples 40 et 41 du document de référence de SKOS. Le premier exemple définit une collection de nom « `MyCollection` » et ayant pour membres la liste de concepts X, Y et Z. Cet exemple est schématisé dans la figure ci-dessous.

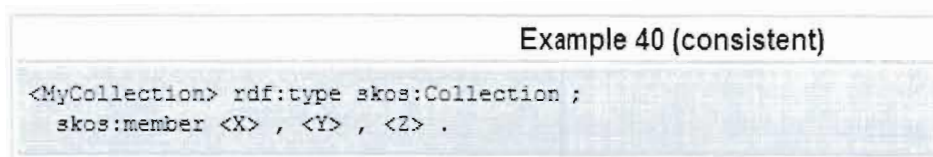


Figure 4.13 - Exemple consistant d'une collection de concepts

Si notre modèle le permet, nous pourrions facilement reproduire cet exemple en utilisant les entités et associations concernées de notre modèle en produisant le diagramme d'objet correspondant. Ce dernier est produit dans le dessin qui suit.

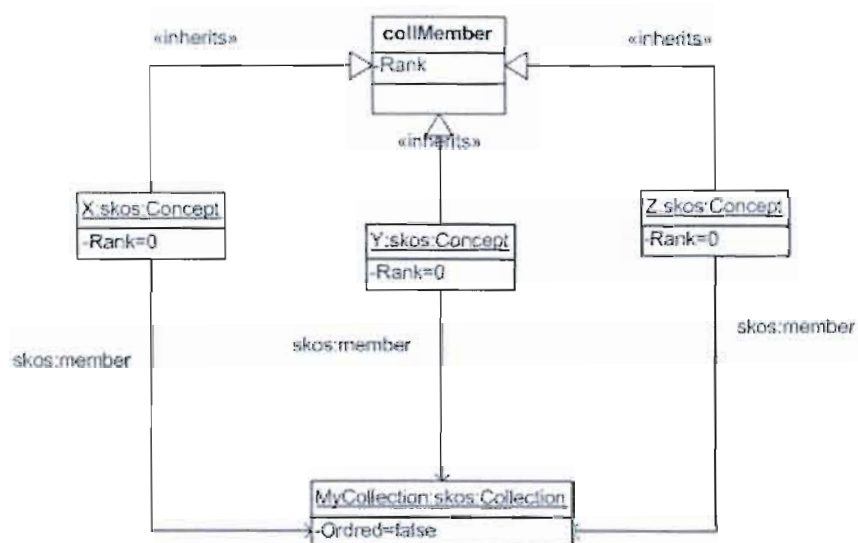


Figure 4.14 - Diagramme d'objet relatif à l'exemple 40

Quant au second exemple, il concerne les collections ordonnées. Dans l'exemple ci-dessous, « `MyOrderedCollection` » est une collection ordonnée qui contient une liste de concepts rangés de 1 à 3 en comptant depuis la gauche.

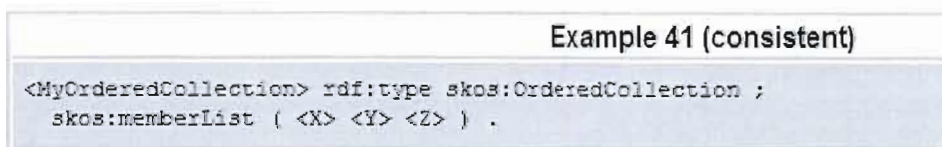


Figure 4.15 - Exemple consistant d'une collection ordonnée de concepts

Cet exemple est supporté par notre modèle conceptuel. En effet, en produisant le diagramme d'objet s'y rapportant, nous avons pu instancier des objets des classes du modèle. Ce diagramme est représenté dans la figure suivante :

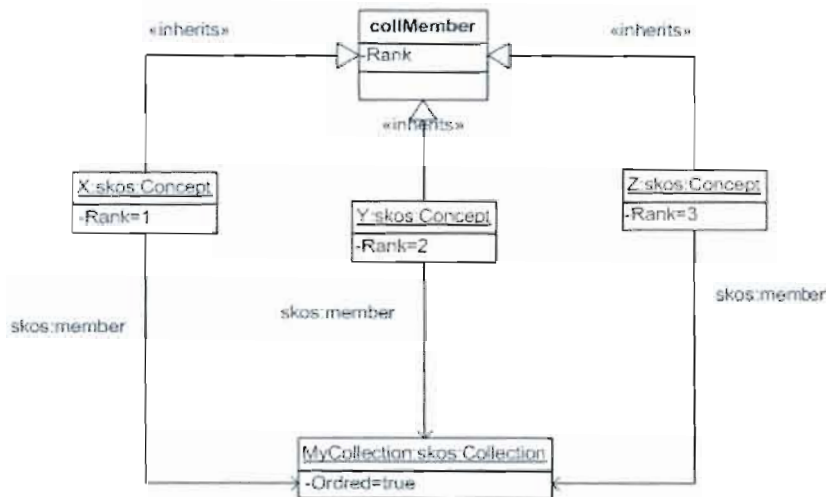


Figure 4.16 - Diagramme d'objet relatif à la sous-propriété « skos:orderedCollection »

4.5.5 Relations sémantiques

Afin de valider la propriété « semanticRelation » de SKOS, nous employons l'exemple 32, représenté dans la figure ci-après, pour vérifier sa compatibilité avec notre modèle UML du standard. Cet exemple nous décrit trois concepts SKOS <A>, et <C> qui ont tous des relations associatives. L'exemple en lui-même a servi à expliquer que même si <A> est associé à et que est associé à <C> ceci ne veut pas dire que <A> est associé à <C>.

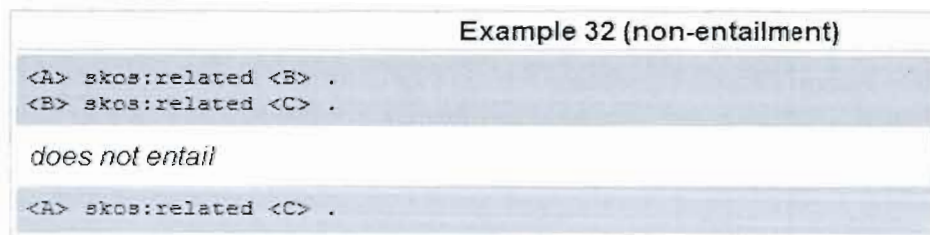


Figure 4.17 - Exemple d'une relation sémantique inconsistante

Dans notre modèle conceptuel de SKOS, la conclusion à laquelle a abouti l'exemple devrait être la même. Ainsi, en instanciant des objets possédant les mêmes valeurs de l'exemple nous sommes arrivés à obtenir les mêmes résultats. En effet, si nous regardons de près les entités représentant les concepts A, B et C, nous pouvons remarquer que les relations « skos:related » existent uniquement entre A et B, et B et C. Par contre aucune relation n'existe entre A et C permettant d'affirmer que A est associé à C. Le schéma qui suit confirme cela.



Figure 4.18 - Diagramme d'objet de l'exemple 32

Bien que l'exemple précédent confirme que cette propriété convient avec le standard SKOS, nous avons effectué un deuxième test. Celui-ci consiste à vérifier que si A est un terme générique de B et que si B est un terme générique de C alors A est un terme générique transitif de B, B est un terme générique transitif de C et A est un terme générique transitif de C. Cet exemple est détaillé dans la figure qui suit.

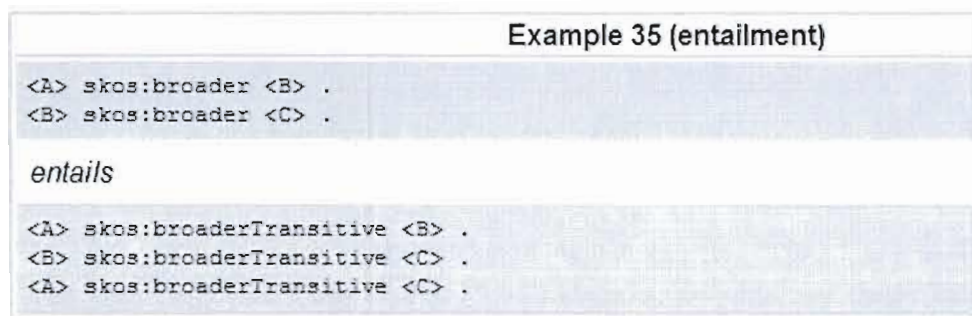


Figure 4.19 - Exemple consistant de relations sémantiques

En se référant au modèle produit de SKOS, nous pouvons instancier les deux premières lignes de code de l'exemple ci-dessus et les schématiser comme suit.

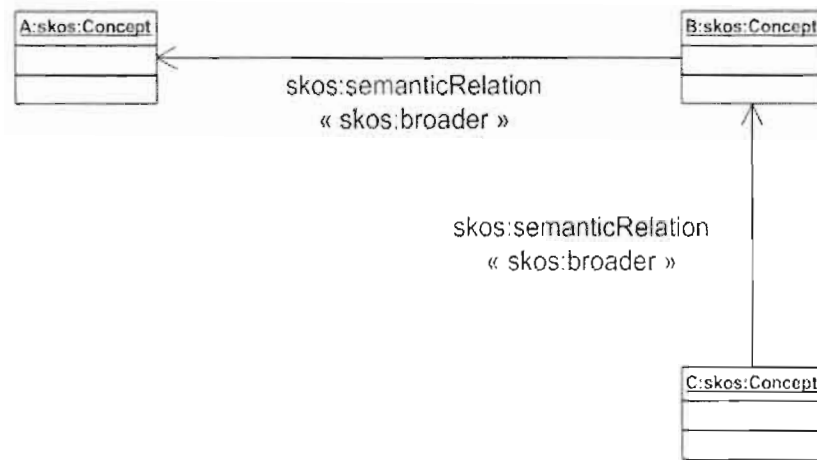


Figure 4.20 - Diagramme d'objet relatif au « skos:broader » de l'exemple 35

Démonstration :

Etant donné que

skos:semanticRelation est une généralisation de skos:broader

Et que

skos:broader est une spécification de skos:broaderTransitive

Alors

skos:broader peut aussi être remplacé par skos:broaderTransitive

On peut alors schématiser comme suit:

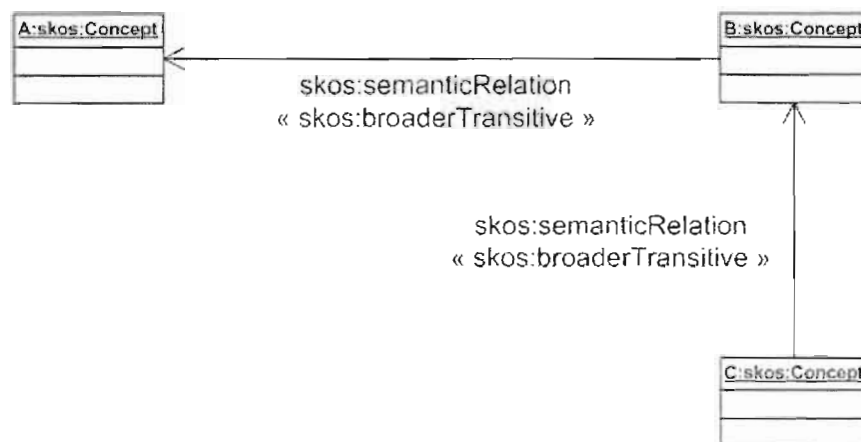


Figure 4.21 - Diagramme d'objet relatif au « skos:broaderTransitive » de l'exemple 35

4.5.5.2 Relations de mapping

Afin de démontrer la capacité de notre modèle à supporter les relations de mapping, nous recourons à l'exemple 56 du document de référence de SKOS. Celui-ci est détaillé dans la figure suivante.

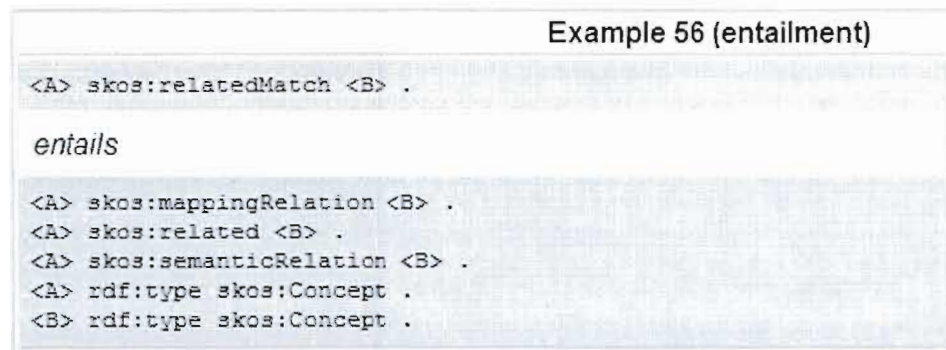


Figure 4.22 - Exemple consistant d'une relation de mapping

Démonstration

En se référant à la figure schématisant le modèle conceptuel de SKOS,

Nous avons skos:relatedMatch est une spécification de skos:mappingRelation et de skos:related

ALORS

<A>skos:mappingRelation

<A>skos:related

Etant donné que skos:mappingRelation et que skos:related sont une spécification de skos:semanticRelation

ALORS

<A>skos:semanticRelation

Donc on aura:

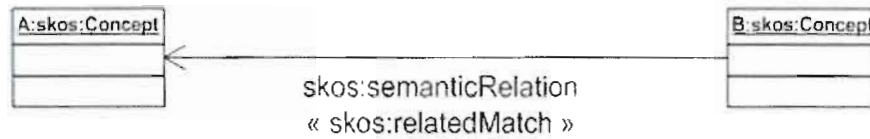


Figure 4.23 - Diagramme d'objet relatif à la sous-propriété « skos:relatedMatch »

Le second exemple, dans la figure ci-après, tend aussi à appuyer nos résultats quant à la propriété « semanticRelation ».

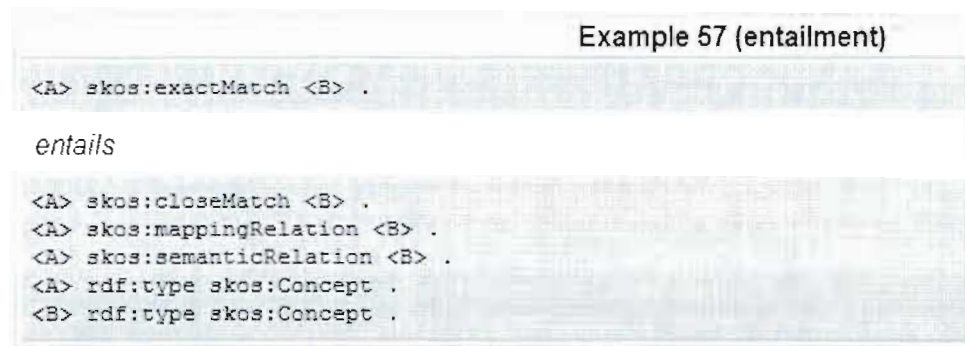


Figure 4.24 - Exemple consistant d'une relation de mapping

Démonstration

Selon le modèle conceptuel que nous avons réalisé,

Nous avons `<A>skos:exactMatch` Et skos:exactMatch est une spécialisation de skos:closeMatch

DONC

`<A>skos:closeMatch`

Etant donné que skos:closeMatch est une spécialisation de skos:mappingRelation

ALORS

`<A>skos:mappingRelation`

Puisque skos:mappingRelation est une spécification de skos:semanticRelation

ALORS

<A>skos:semanticRelation

ET comme skos :semanticRelation est une relation entre deux concepts alors on peut dire que

<A>skos :Concept et skos :Concept

Et on aura:

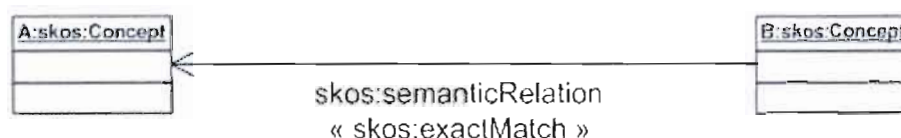


Figure 4.25 - Diagramme d'objet relatif à la sous-propriété « skos:exactMatch »

4.5.6 Étude de cas de l'observatoire international des coopératives de services financiers

L'observatoire international des coopératives de services financiers (2010) est une organisation à but non lucratif qui a pour mission de contribuer au développement de l'analyse scientifique des coopératives de services financiers et leur milieu en donnant aux chercheurs universitaires et institutionnels accès aux connaissances existantes et aux données pertinentes.

Le projet actuel qu'il réalise consiste, en une partie, à réaliser un thésaurus interopérable contenant les acronymes des institutions financières. Étant donné que notre projet de recherche couvre pleinement cette partie de leur travail, nous sommes entrés en contact avec eux afin de valider, d'une part, une partie de notre modèle conceptuel de SKOS et de les aider, d'autre part, à réaliser ce thésaurus par l'intermédiaire de notre outil.

Ainsi pour valider une partie de notre modèle conceptuel de SKOS, nous avons retenu des exemples d'acronymes de la liste procurée par l'observatoire international des coopératives. A partir de cette liste nous produisons des diagrammes d'objets correspondant à chacun des exemples afin de vérifier la validité de notre modèle.

Terme retenu	Langue	Synonyme(s) (intra langue)	Synonyme(s) (interlangue)
Banco Credicoop Cooperativo Limitado	Espagnol	Banco Credicoop C.L. Banco Credicoop COOP. LTDO.	
Bancoob	Portugais	Banco Cooperativo do Brasil S/A Banco Cooperativo do Brasil	
BVR	Allemand	Bundesverband der Deutschen Volksbanken und Raiffeisenbanken	Federal Association of German Cooperative Banks

Tableau 4.1 - Echantillon des acronymes de l'observatoire international des coopératives

Pour réaliser cette vérification, nous avons trouvé des difficultés concernant les termes employés pour décrire le contenu de ce tableau. En effet, certaines ambiguïtés subsistent dans le tableau notamment concernant la signification donnée au terme retenu. Pour l'observatoire des coopératives, un terme retenu correspondrait soit à un acronyme et les synonymes à leur signification, soit à un terme préféré et les synonymes à des acronymes ou à leur traduction dans une autre langue.

Dans les différents cas, il existe principalement deux manières de représenter ces termes retenus dans le standard SKOS. En effet, si on considère que le terme retenu correspond à un terme préféré alors nous le représentons, dans SKOS, en créant un nouveau concept ayant pour label préféré le terme retenu et pour label alternatif les synonymes inter et intra langue. La figure suivante synthétise la première solution.

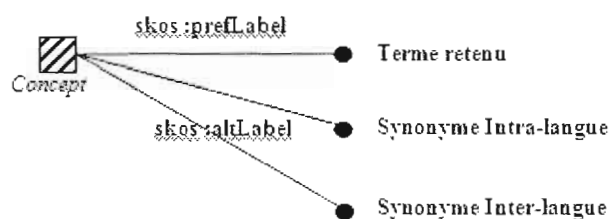


Figure 4.26 - Proposition d'utilisation du label SKOS

Nous utilisons deux exemples du tableau fourni par l'observatoire pour illustrer cette première solution. La figure ci-dessous représente le diagramme d'objet de notre premier exemple.

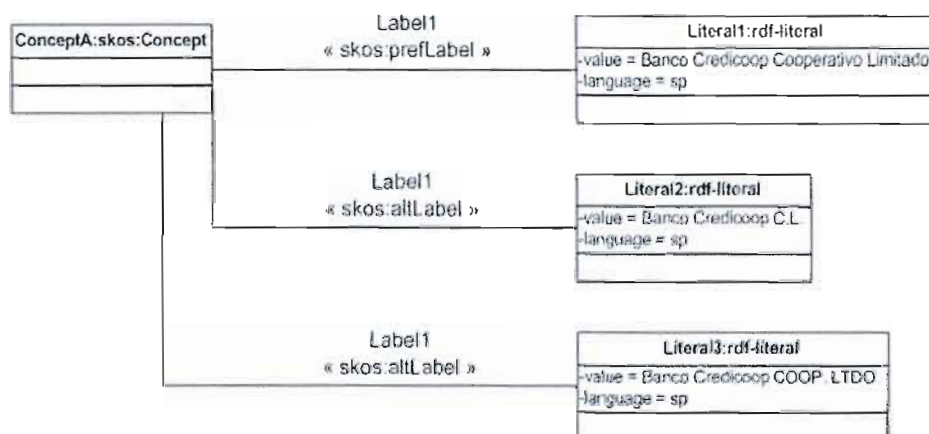


Figure 4.27 – Exemple du « Banco Credicoop » illustrant la première solution

Ci-après, une représentation du diagramme d'objet du deuxième exemple appliqué suivant la première solution.

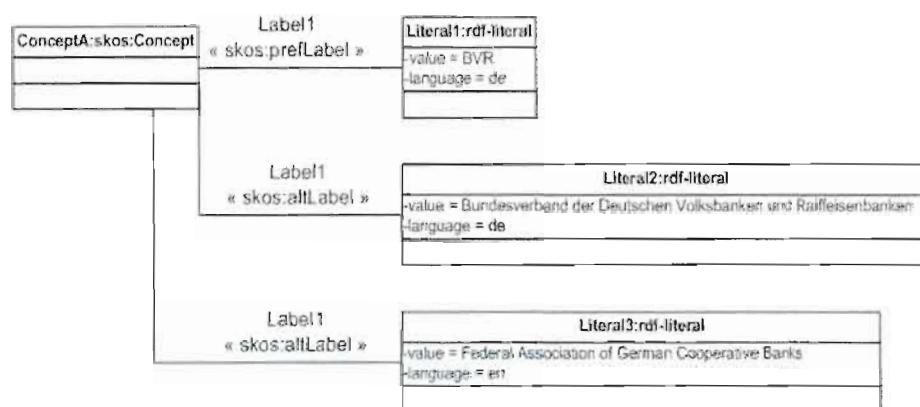


Figure 4.28 - Exemple du « BVR » illustrant la première solution

Aussi pourrions-nous représenter le terme retenu en recourant à l'extension de SKOS. Grâce à SKOS-XL, nous pouvons créer un seul concept ayant plusieurs labels pour contenir le terme relatif ainsi que les synonymes. Ces labels sont ensuite reliés grâce à des relations spécifiques telles que, dans notre cas, les synonymes et les acronymes. Le dessin ci-après schématise cette solution.

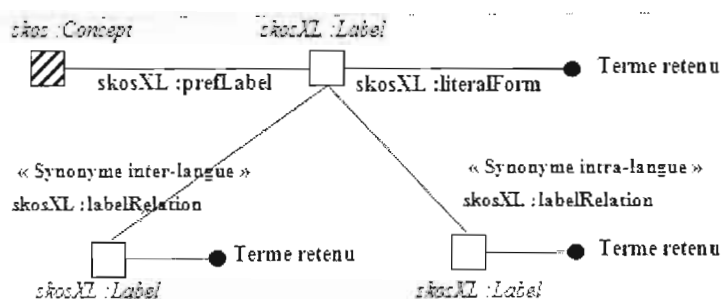


Figure 4.29 - Proposition de représentation en utilisant SKOS-XL

Pour illustrer cette seconde solution, nous utilisons les mêmes exemples que pour l'explication de la première. Nousinstancions le premier exemple dans la figure qui suit en recourant encore au diagramme d'objet.

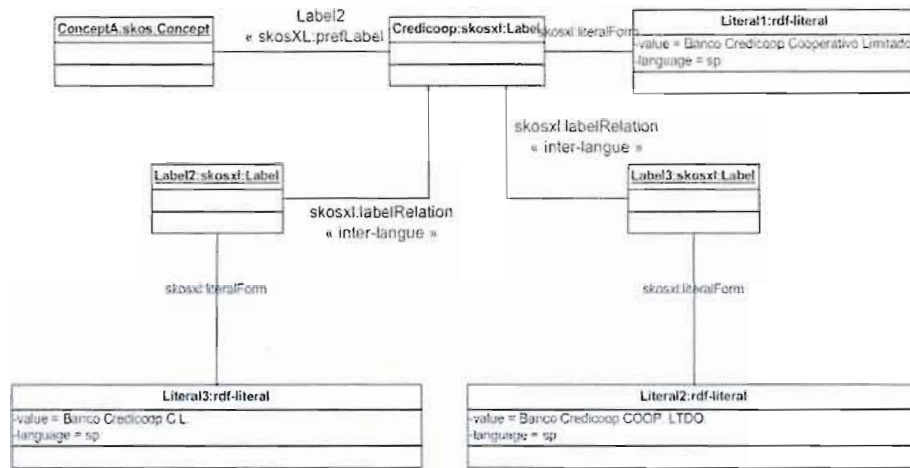


Figure 4.30 - Exemple du « Banco Credicoop » illustrant la deuxième solution

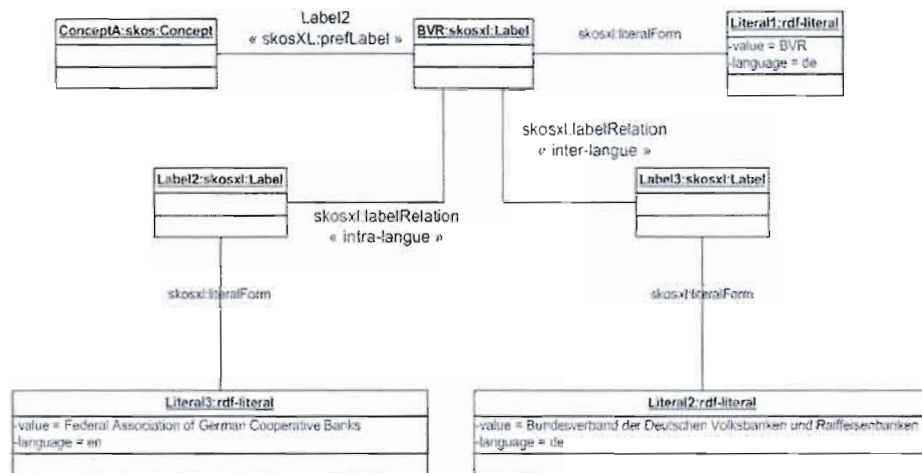


Figure 4.31 - Exemple du « BVR » illustrant la deuxième solution

Comme nous avons pu le démontrer grâce aux diagrammes d'objet instanciant deux exemples du tableau fourni par l'observatoire international des coopératives, les deux solutions sont supportées par notre modèle conceptuel de SKOS.

Les deux diagrammes d'objets démontrent que non seulement le modèle soutient ces deux solutions mais aussi que les deux solutions fournissent le même résultat. Le choix de

l'une et de l'autre solution dépendra de nos préférences lors de la phase de développement de notre outil. Cependant, pour permettre un bon emploi de notre outil et une meilleure flexibilité d'utilisation, nous envisageons d'utiliser la deuxième solution.

4.6. CONCLUSION

Après avoir effectué les modifications nécessaires et avoir validé le méta-modèle SKOS, ce dernier constitue désormais une base solide pour le développement de l'outil logiciel que nous comptons réaliser. Grâce à l'étape de validation du méta-modèle, nous avons pu raffiner et améliorer notre modèle conceptuel ainsi que son schéma de base de données. Désormais, nous sommes persuadés que le méta-modèle que nous proposons est conforme dans son intégralité au méta-modèle théorique présenté par le W3C.

A l'avenir, nous ne nous soucions plus du méta-modèle SKOS et nous nous concentrerons plus sur la conception et le développement de l'outil logiciel SKOS. Celui-ci fera l'objet du prochain chapitre.

.

CHAPITRE V

LE SYSTÈME LOGICIEL

5.1. INTRODUCTION

Après avoir validé le modèle conceptuel de SKOS ainsi que le schéma de la base de données qui le supporte, nous passons à la réalisation de l'outil logiciel. Pour ce faire, nous utiliserons le schéma de la base de données pour la construction du noyau de l'outil qui constitue l'élément central pour le stockage des données sous SKOS et pour le bon fonctionnement de l'outil en lui-même. Une fois celle-ci achevée, nous entamerons la phase de réalisation. Lors de cette étape, nous discuterons en premier lieu des règles de correspondance entre le modèle conceptuel de SKOS et ceux des systèmes d'organisation des connaissances qui sont fondamentaux pour le développement de l'outil.

Mais tout d'abord, nous commençons, dans la section suivante, par présenter l'outil logiciel que nous réaliserons lors de ce travail de recherche.

5.2. PRESENTATION DE L'OUTIL LOGICIEL

Afin d'avoir une idée globale sur l'outil et de faciliter la compréhension des tâches accomplies pour son développement, nous décrivons dans ce qui suit l'architecture physique de l'outil et nous survolerons ses différentes fonctionnalités pour finir par présenter les technologies utilisées pour l'implémentation de l'outil ainsi que sa base de données.

5.2.1 Architecture

L'outil de gestion des ensembles de données terminologiques basé sur SKOS s'articule autour de trois principaux axes : la base de données représentative de la norme SKOS, le

mécanisme de « mapping » et les modules correspondant aux fonctionnalités de l'outil. Le schéma ci-dessous décrit l'architecture physique de l'outil et l'interaction de ses différents éléments.

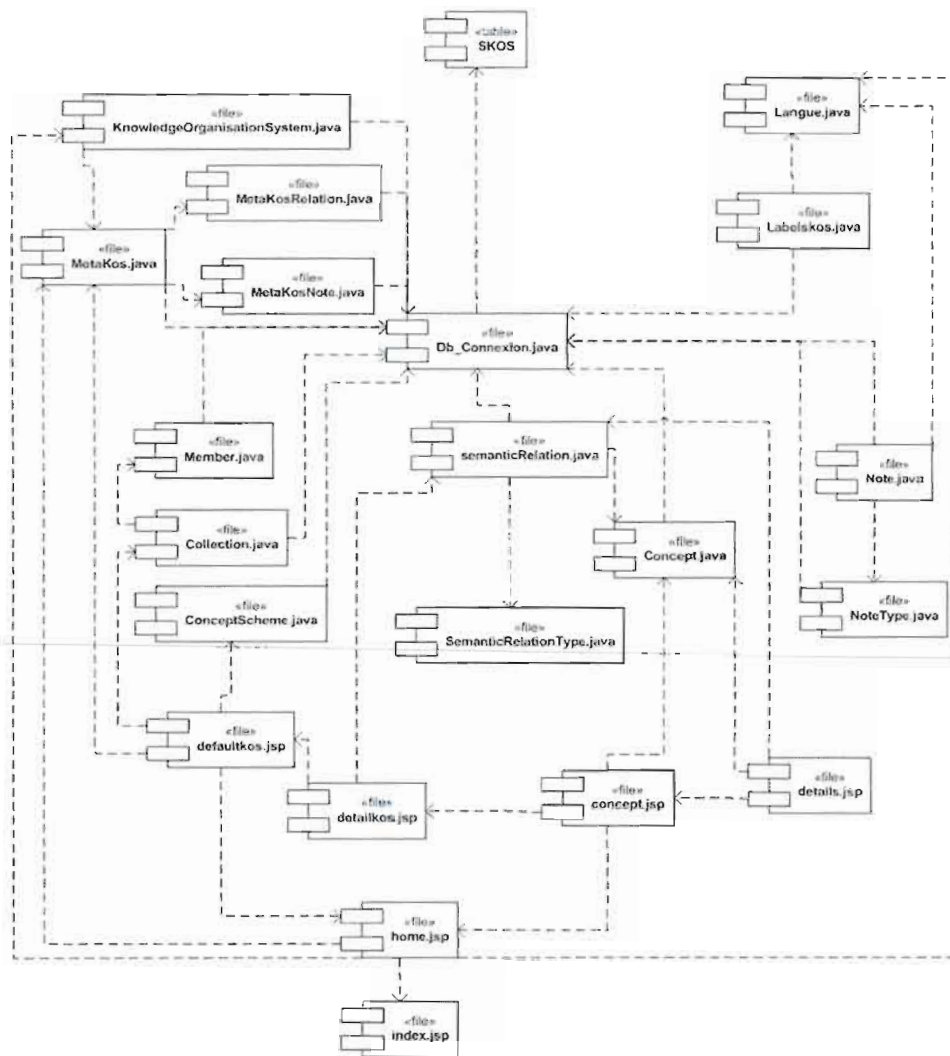


Figure 5.1 - Architecture physique de l'outil de gestion des ensembles de données terminologiques basée sur SKOS

Cette architecture repose essentiellement sur des fichiers Java et Jsp. Les fichiers Java sont utilisés pour représenter les principales propriétés de SKOS comme les concepts, les collections et les relations sémantiques, et à gérer le stockage de leurs données dans la

BD, alors que les Jsp sont employés pour réaliser des traitements à partir d'instanciation d'objets des classes Java.

Il est à noter que nous employons une classe particulière « Db_Connexion.java » qui sert de point d'accès entre la base de données SKOS et les classes Java et ce, pour préserver la sécurité des données et pour optimiser le code.

5.2.2 Fonctionnalités du logiciel

L'outil « SKOS TOOL » que nous présentons dans ce chapitre, a pour principal but la gestion d'ensembles de données terminologiques. Il offre donc des fonctionnalités de gestion dont la création, la modification et la suppression pour des systèmes d'organisation des connaissances dont trois en particulier implémentés par défaut : glossaires, thésaurus et glossaurus, qui combine les propriétés d'un thésaurus et d'un glossaire. Cependant, étant donné que notre outil tend à gérer toute variété de KOS, nous proposons une fonction qui permet aux utilisateurs de créer leurs ensembles et ce, en convenance avec les propriétés de SKOS. Il est à noter que cette fonctionnalité ne sera disponible que pour un type spécifique d'utilisateurs. En effet, l'outil de gestion de KOS s'adresse principalement à deux catégories d'utilisateurs : les administrateurs et les auteurs.

Les auteurs qui sont les utilisateurs basiques de cet outil, ont accès à certaines fonctionnalités. Celles-ci sont principalement liées à la gestion des KOS déjà implantés dans l'outil tel que les thésaurus, les glossaires et les glossaurus ou développés par l'administrateur. L'outil leur permet aussi d'exporter des ensembles de données terminologiques normalisés sous SKOS en un format XML exploitable par d'autres applications.

Les administrateurs qui sont des supers utilisateurs ont eux accès à la totalité des fonctionnalités de l'outil. En plus de pouvoir gérer les ensembles de données terminologiques basiques, ils leur sont permis de créer de nouvelles formes de KOS. En effet, comme notre outil propose que trois types de KOS (glossaire, thésaurus et glossaurus), nous proposons aux administrateurs de construire leur propre forme de systèmes de gestion des connaissances. Par ailleurs, les administrateurs ont le pouvoir de

gérer les concepts SKOS directement sans pour autant les chercher dans leurs ensembles de données terminologiques.

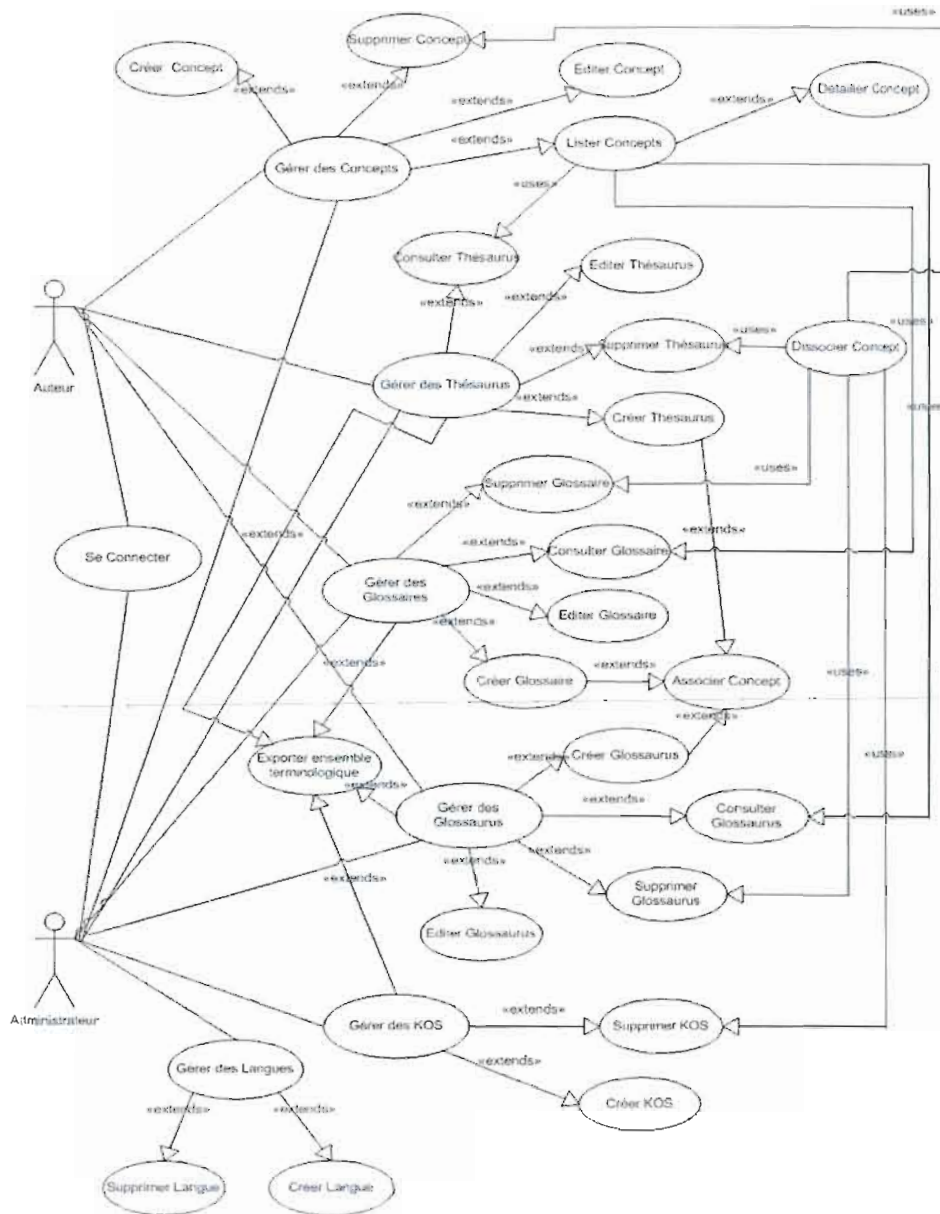


Figure 5.2 - Diagramme de cas d'utilisation de « SKOS TOOL »

5.2.3 Environnement et langages de développement

La réalisation de notre outil de gestion des données terminologiques basé sur SKOS n'aurait pas pu se faire sans le recours à différents environnements et langages de développement.

5.2.3.1 MySQL

MySQL Server (Oracle, 2010) est un système de gestion de bases de données relationnelle. Il fonctionne en utilisant une architecture client/serveur dans laquelle le serveur s'exécute sur la machine contenant les bases de données et les clients, les programmes qui se connectent au serveur de base de données en émettant des requêtes dans un format préétabli.

C'est un système de bases de données relativement simple et très performant. Grâce à son GUI « MySQL Workbench », il est facile de configurer, de surveiller et de gérer une base de données. Aussi cet outil propose-t-il des fonctionnalités telles que la production de modèle entité-association à partir d'une base de données ou l'inverse.

5.2.3.2 L'EDI « Netbeans 6.8 »

Netbeans 6.8 (2009) est un environnement de développement intégré « EDI » supporté par SUN Microsystems. Il est généralement employé pour l'implémentation de projets en différents langages de programmation et plus spécifiquement dans le développement Java. Il est particulièrement bien adapté au développement d'applications WEB.

Netbeans offre la possibilité de développer des projets multi-langages de différents types et aide les développeurs dans la réalisation de leurs tâches en mettant à leur disposition des éditeurs graphiques d'interfaces. Par ailleurs, il offre une documentation riche et détaillée facilitant ainsi son utilisation.

Les palettes d'outils dont il dispose permettent de construire différents composants Java EE tels que les Enterprise Java Beans « EJBs » et les servlets, d'indiquer des erreurs et proposent un débogueur permettant d'effectuer une exécution pas à pas afin d'en venir à bout.

5.2.3.3 Java Server Pages ou JSP

Les Java Server Pages sont un standard qui permet de développer des applications web interactives. C'est un langage de script, comparable aux scripts PHP, ASP et CGI et qui est exécuté du côté serveur (A. Tasso et al., 2004)

Ils permettent de générer dynamiquement des pages web grâce à l'utilisation de balises JSP repéré par les symboles `<%` et `%>`. Ces derniers permettent au serveur web d'interpréter du code Java qui y est contenu afin de renvoyer du code HTML au navigateur du client. Ce code renvoyé vient, par la suite, s'insérer dans le contenu web statique HTML/XHTML afin de former la page « JSP ».

5.2.3.4 XML

XML ou « eXtensible Markup Language » (T. Bray et J. Paoli, 1998) est une des recommandations qui ont été émises par le W3C. C'est un langage de balisage « qui a pour vocation d'être un format d'échange de documents entre diverses applications » (J. Euzenat, 2000). Il permet de structurer et de formater les documents grâce à des balises définies par ses utilisateurs. C'est en quelque sorte un métalangage.

La facilité d'usage de XML, son universalité et sa portabilité sont quelques propriétés parmi d'autres qui sont captivantes. En ce qui concerne notre projet de recherche, l'un de nos objectifs de recherche, cité plus haut, concerne l'interopérabilité des ensembles de données terminologiques, qui sous entend leur portabilité.

Certes, SKOS permet de représenter ces ensembles sous une forme normalisée et prête à être transportée. Cependant, il n'offre pas de moyen de les porter par lui-même. Pour s'y faire, il recourt au langage XML.

5.3. CONSTRUCTION DE LA BASE DE DONNEES

Après avoir présenté l'environnement et les langages de développement, nous déployons, dans ce qui suit, la base de données depuis le modèle conceptuel que nous avons réalisé et décrit dans le chapitre précédent. Mais tout d'abord, nous commençons par présenter le logiciel ROBOT qui servira à la construction de notre BD.

5.3.1 Le logiciel « ROBOT »

« ROBOT » est un logiciel qui a été réalisé par le Professeur Olivier Gerbé de HEC Montréal. Cet outil permet, à partir d'un modèle « entité-association », de créer des bases de données relationnelles et de générer le code permettant leur présentation à l'utilisateur et leur manipulation. Son intérêt réside dans la simplicité offerte à l'utilisateur pour créer des bases de données, administrer leurs tables, les remplir ou les modifier.

Le logiciel « ROBOT » est constitué de deux ensembles distincts. Le premier appelé « domaine » comprend les tables de la nouvelle base de données. Si l'on considère l'exemple schématisé dans la figure ci-dessous représentant une mini base de données, le domaine sera constitué des deux entités « concept », « conceptscheme » et de l'association « inscheme ».

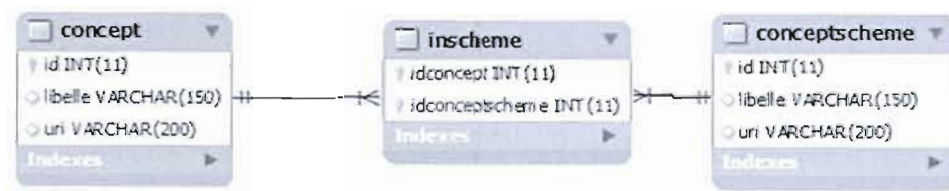


Figure 5.3 - Diagramme de classe d'un modèle « entité-association »

Quant au second, nommé « Robot », il comprend six tables : « Type », « Attribut », « Présentation », « lien », « relation » et « Élément de présentation », qui sont responsables de la création des vues présentées à l'utilisateur et il leur permet de gérer les données qui sont enregistrées dans la partie « Domaine ». Dans notre exemple, cette vue sera présentée comme suit :

La figure montre deux interfaces utilisateur pour la gestion d'une base de données. À gauche, l'interface pour la table **concept** permet de saisir le libelle (exemple : "être vivant"), l'URI (exemple : "http://www.exemple-ci-t"), et de gérer des prélabels. À droite, l'interface pour la table **conceptscheme** permet de sélectionner des catégories, ici "Biologie" et "Géologie".

concept

Libelle : être vivant

URI : http://www.exemple-ci-t

Prélabel :

× tr Libelle Être vivant

[Ajouter un prélabel](#)

ConceptScheme :

[Insérer dans un schéma de concept](#)

conceptscheme

☐ Sélectionner tout

☒ Biologie

☐ Géologie

Figure 5.4 - Vue présentée à l'utilisateur de la base de données

Dans notre projet de recherche, nous nous désintéresserons de cette partie car nos objectifs de recherche diffèrent de ceux de la partie « Robot ». Par conséquent, nous nous arrêterons à l'utilisation de la première partie pour insérer notre modèle conceptuel de SKOS et ainsi construire notre BD.

5.3.2 Insertion de la base de données SKOS

Afin de réaliser notre base de données SKOS, nous procédons par différentes étapes. Tout d'abord, nous commençons par définir le « domaine » de « ROBOT ». Pour ce faire, nous représentons notre modèle conceptuel de SKOS en XML en définissant des balises qui seront, par la suite, interprétées par « ROBOT ».

Les balises utilisées par « ROBOT » sont nombreuses et variées. Nous présentons les plus importantes d'entre elles :

- `<table></table>` définit une table. Sans cette balise, la création d'une entité ou d'une association est impossible.
- `<attribute></attribute>` définit les attributs d'une table.
- `<name></name>` associe un nom à une table ou un attribut.
- `<type></type>` identifie le type d'un attribut, d'une table. Il est à noter que « ROBOT » différencie une entité d'une association. Pour les distinguer on associe au type la valeur « entity » ou « relation ».
- `<primkey></primkey>` définit un attribut comme étant une clé primaire.

Ci-dessous, un fragment de code XML représentant l'entité « concept » de la figure 5.3.

```

<?xml version="1.0" encoding="UTF-8" ?>
<model code="SKOS2">
  <!-- CONCEPT -->
  <table>
    <name>concept</name>
    <affname>Concept</affname>
    <type>entity</type>
    <active>Y</active>
    <attribute>
      <name>id</name>
      <affname>id</affname>
      <type>int</type>
      <size>11</size>
      <primkey>Y</primkey>
    </attribute>
    <attribute>
      <name>libelle</name>
      <affname>Libelle</affname>
      <type>varchar</type>
      <size>150</size>
      <primkey>N</primkey>
    </attribute>
    <attribute>
      <name>uri</name>
      <affname>URI</affname>
      <type>varchar</type>
      <size>200</size>
      <primkey>N</primkey>
    </attribute>
  </table>

```

Figure 5.5 - Représentation XML de la table « concept » de l'entité Concept du modèle conceptuel de SKOS

Dans cet exemple, nous avons une représentation de l'entité concept identifiée par les balises `<table></table>` possédant trois différents attributs : id, libelle, URI identifiés par les balises `<attribute> </attribute>` et contenant des propriétés telles que son nom, son type (varchar, int, text, etc.), sa taille et une clé primaire pouvant avoir comme valeur « Y » ou « N » pour affirmer ou infirmer si l'attribut est une clé primaire. Nous procédons de la même manière avec toutes les entités du modèle conceptuel de SKOS en effectuant les modifications nécessaires.

Une fois toutes les entités et associations définies, nous créons les liaisons entre elles. Pour ce faire, nous utilisons les balises suivantes :

- `<relation></relation>` définit une relation entre une entité et une association.

- `<link></link>` identifie un lien entre un attribut d'une entité et celui d'une association.
- `<linktablekey></linktablekey>` définit l'attribut de l'association à référencer.
- `<reftablekey></reftablekey>` définit l'attribut de référence dans l'entité à lier.

Dans le cas de notre exemple cité plus haut, ceci donnera le code XML suivant :

```
- <relation>
  <name>inscheme-concept</name>
  - <link>
    <linktablekey>inscheme.idconcept</linktablekey>
    <reftablekey>concept.id</reftablekey>
  </link>
</relation>
- <relation>
  <name>inscheme-conceptscheme</name>
  - <link>
    <linktablekey>inscheme.idconceptscheme</linktablekey>
    <reftablekey>conceptscheme.id</reftablekey>
  </link>
</relation>
```

Figure 5.6 - Représentation XML de la relation entre « concept - inscheme » et celle entre « conceptscheme - inscheme »

Ensuite, dès que le code XML est finalisé, nous exécutons un script de transformation XSLT¹⁰ sur ce fichier afin de produire un fichier SQL. Ces scripts sont exécutés en utilisant la commande « C:\...\MSXML 4.0\msxsl.exe » et en spécifiant les trois paramètres qui suivent :

- \$BaseName.xml ../XSLT/robot2schema.xsl -o ../SQL/\$BaseName-Schema.sql pour produire le schéma de la base de données SKOS.
- \$BaseName.xml ../XSLT/robot2sql.xsl -o ../SQL/\$BaseName-Robot.sql
- \$BaseName.xml ../XSLT/robotdata2sql.xsl -o ../SQL/\$BaseName-Data.sql afin d'initialiser la base de données.

¹⁰ XSLT : eXtensible Stylesheet Language Transformations

Finalement, les fichiers SQL produits suite à l'exécution des scripts XSLT sont importés dans MySQL et exécutés afin de créer la base de données finale qui sera utilisée comme espace de stockage de notre engin SKOS.

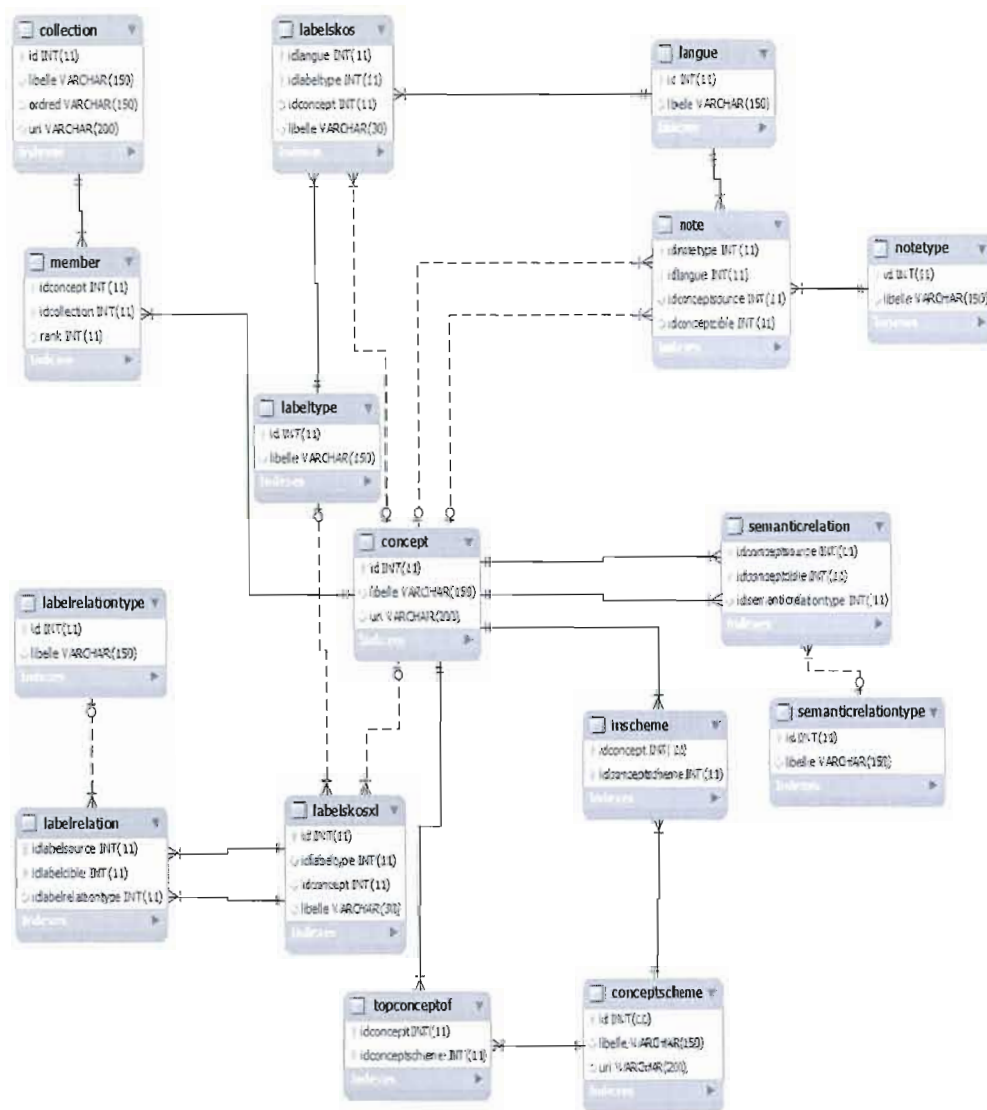


Figure 5.7 – Modèle entité-association de SKOS

La base de données obtenue subit par la suite une modification finale pour intégrer des tables métiers afin de permettre la réalisation de certaines fonctionnalités dont la création de nouveaux systèmes de gestion des connaissances tels que les schémas de classification

ou les ontologies en respectant d'une part le standard SKOS et d'autre part la terminologie utilisée pour nommer leurs propriétés. Certaines entités du modèle conceptuel de SKOS telles que « conceptscheme » ou « collection » ont été affectées par ces modifications. Une clé étrangère leur a été greffée pour pouvoir identifier les méta-informations qui les concernent.

La figure suivante présente les entités qui sont responsables de l'identification des propriétés des systèmes de gestion des connaissances. Ces tables intègrent par défaut les propriétés concernant les glossaires et les thésaurus.

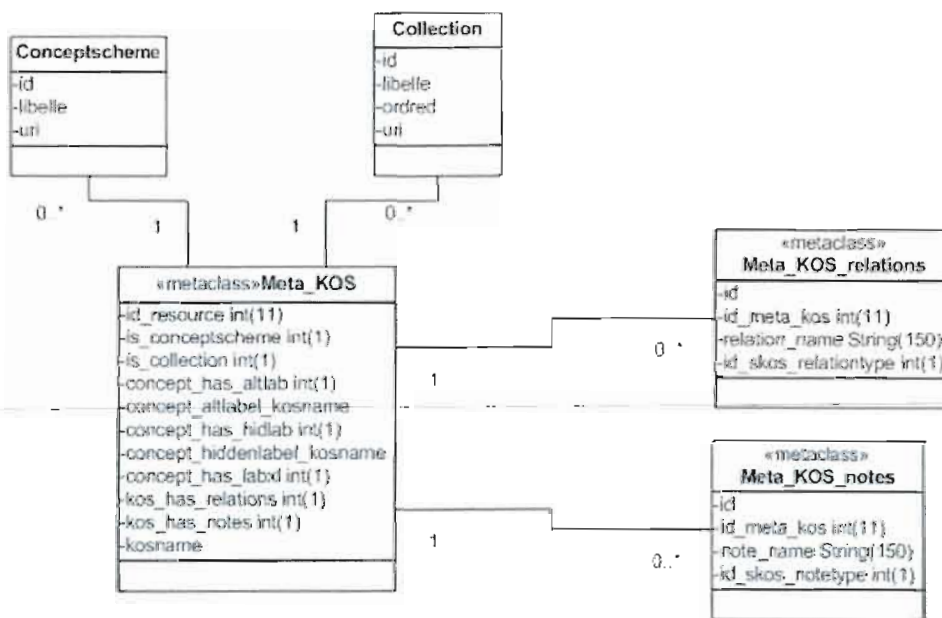


Figure 5.8 - Représentation conceptuel de tables métiers correspondant à la fonctionnalité de gestion des KOS

5.4. DEVELOPPEMENT DE SKOS TOOL

Après avoir établi la base de données, à partir du modèle conceptuel qui a été validé lors du chapitre précédent, nous développons désormais l'outil qui le supporte.

Cet outil doit respecter certaines contraintes et besoins. En effet, l'application doit permettre une utilisation facile, simple et intuitive. Il doit, en outre, présenter les ensembles terminologiques sous leur forme naturelle. Par exemple, pour un glossaire, sa

présentation générale est souvent sous forme de suite de termes auxquels sont associés des définitions. Cette représentation devrait être la même que dans l'outil que nous développons. De même pour la présentation des autres formes de système de gestion des connaissances. L'outil doit aussi présenter certaines fonctionnalités que nous décrivons dans la section suivante.

5.4.1 Les fonctionnalités du système logiciel

Les fonctionnalités du système logiciel que nous présentons dans cette section sont destinées à deux types d'utilisateurs. Les fonctions décrites de la section 5.4.1.1 jusqu'à la section 5.4.1.4 sont destinées aux deux sortes d'utilisateurs. Quant aux restes des fonctionnalités, elles sont assignées aux administrateurs.

5.4.1.1 Gestion des glossaires

L'outil de gestion des ensembles de données terminologiques renferme une fonction de gestion des glossaires. Cette dernière permet de créer, d'éditer et de supprimer des glossaires. Elle permet aussi de sélectionner, depuis une liste, un glossaire afin de visualiser les termes qu'il contient.

Afin de créer ou d'éditer un glossaire, l'outil propose à l'utilisateur une interface simple lui permettant de saisir des informations telles que le nom du glossaire. Une fois les informations introduites et validées, l'outil les enregistre en recourant au mécanisme de mapping entre les propriétés du KOS et celui du méta-modèle SKOS. Ces règles de correspondances ont été préalablement établies et font l'objet d'explication dans la section 4.5.1. La figure ci-dessous donne un aperçu d'une de ces règles.

The screenshot shows a web interface for managing glossaries. It consists of three main parts:

- Nouveau glossaire form:** Contains fields for 'Nom du glossaire' and 'URI'. Below these fields are two buttons: 'Enregistrer' (Register) and 'Annuler' (Cancel).
- collection table:** A table with the following fields: 'id INT(11)', 'libelle VARCHAR(150)', 'ordred TINYINT(1)', 'uri VARCHAR(200)', and 'idmetakos INT(11)'. It has an 'Indexes' link at the bottom.
- metakos table:** A table with the following fields: 'id INT(11)', 'is_conceptscheme TINYINT(1)', 'is_collection TINYINT(1)', 'concept_has_alttab TINYINT(1)', 'concepts_alttab_kosname VARCHAR(150)', 'concept_has_preftab TINYINT(1)', 'concepts_hiddenlab_kosname VARCHAR(150)', 'kos_has_relations TINYINT(1)', 'kos_has_notes TINYINT(1)', and 'kosname VARCHAR(150)'. It also has an 'Indexes' link at the bottom.

Arrows indicate the following relationships:

- An arrow points from the 'Nom du glossaire' field to the 'libelle' field in the 'collection' table.
- An arrow points from the 'URI' field to the 'uri' field in the 'collection' table.
- An arrow points from the 'art contemporain' text input to the 'libelle' field in the 'collection' table.
- An arrow points from the 'art contemporain' text input to the 'art_contemporain' field in the 'metakos' table.
- A dashed arrow points from the 'idmetakos' field in the 'collection' table to the 'id' field in the 'metakos' table.

Figure 5.9 - Création et édition d'un glossaire

Outre ces opérations, la gestion des glossaires permet aussi de gérer les termes qu'il contient. Il propose entre autres soit d'ajouter un nouveau terme soit d'en associer ou dissocier un.

Lorsqu'un utilisateur ajoute un nouveau terme, l'interface ci-dessous à gauche (figure 5.9) apparaît. Il lui revient alors de remplir les différents champs sans quoi l'enregistrement ne pourra pas se faire. Une fois les champs renseignés, la sauvegarde peut se faire et le mécanisme de correspondance démarre. Ce mécanisme reste invisible pour l'utilisateur. Il a l'impression d'interagir avec une interface ordinaire d'un gestionnaire de glossaires. Notons que le type de glossaire implémenté par défaut dans notre outil est un glossaire multilingue permettant d'associer à chaque terme un exemple.

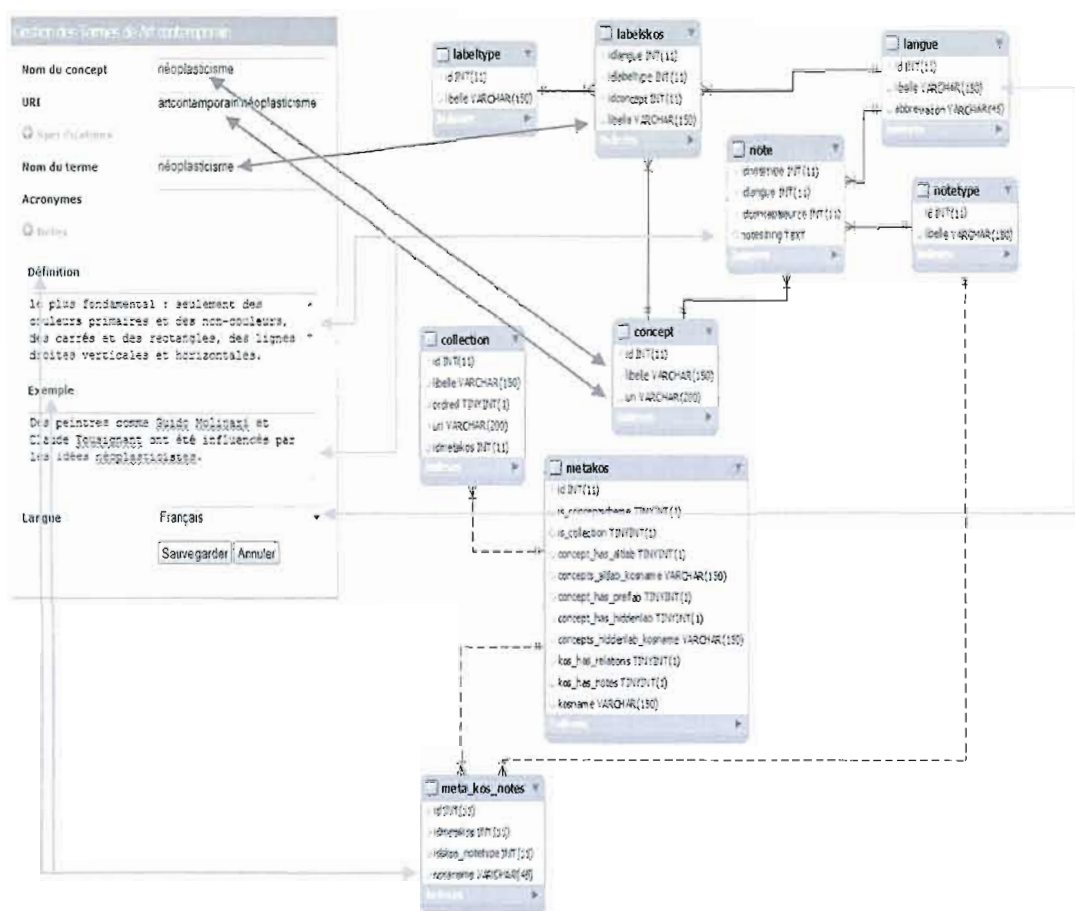


Figure 5.10 - Création d'un nouveau terme d'un glossaire

La gestion des termes d'un glossaire implique aussi l'affichage de tous leurs attributs. Le libellé d'un terme, tout comme sa définition, est affiché en une ou plusieurs langues. Ces attributs peuvent faire l'objet d'ajout, de suppression, ou de modification. Il est à noter que la suppression n'est pas permise lorsqu'un terme ne possède qu'un seul nom préféré.

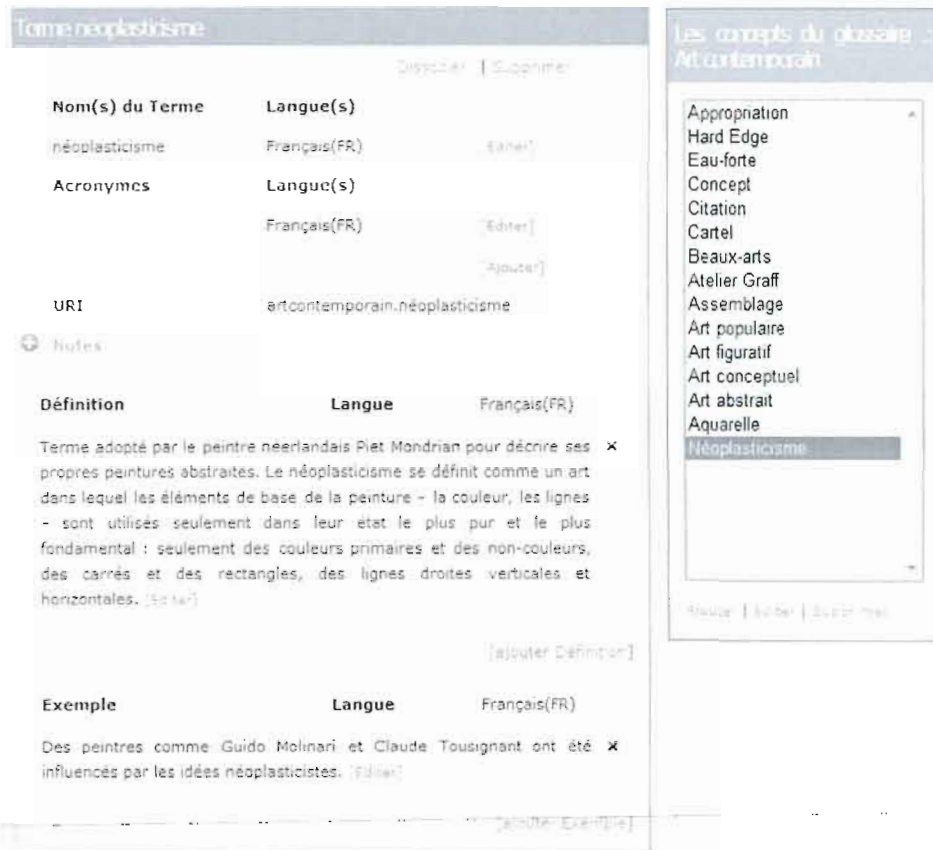


Figure 5.11 - Affichage détaillé d'un terme d'un glossaire

5.4.1.2 Gestion des thésaurus

La gestion des thésaurus offre le même type d'opération que la gestion des glossaires. En effet, cette fonctionnalité permet de représenter un thésaurus dans sa forme naturelle tout en recourant au mécanisme de « mapping » que nous avons développé. La gestion des thésaurus permet, en plus de la visualisation des différents termes qu'un thésaurus contient, de spécifier les diverses relations qui existent entre eux.

Tout comme les glossaires, si un thésaurus ne possède aucun terme, l'outil suggère d'en créer ou d'en associer un ou plusieurs déjà existants. L'outil offre aussi d'ajouter les différentes relations qu'un thésaurus peut avoir. L'utilisateur peut, par exemple, ajouter une relation de type « TS – Terme Spécifique » en associant un terme à la liste des termes

spécifiques. Les autres sortes de relations, que nous avons développées plus haut, sont aussi supportées par notre outil.

Terme Loi 1987 (emploi des personnes handicapées) Dissocier | Supprimer

Nom(s) du Terme	Langue(s)	
Loi 1987 (emploi des personnes handicapées)	Français(FR)	Editer Ajouter

URI LawCongress.Loi_1987

Relation Générique (TG)

- Loi 1975 (handicap) ×
- Loi 2005 (handicap) ×

Relation Spécifique (TS)

Aucun terme n'a été attaché jusqu'à présent. Voulez-vous Ajouter une nouvelle relation?

Termes associés « TA »

Termes "Employé Pour" « EP »

Les concepts du thésaurus : Law Congress

- Loi 1975 (handicap)
- Loi 1987 (emploi des persoi
- Loi 1988 (recherche bioméd
- Loi 2005 (handicap)
- Loi 2005 (droits des malade

Ajouter Associer Editer

Figure 5.12 - Affichage d'un terme du thésaurus « Law Congress »

5.4.1.3 Gestion des glossaurus

Un glossaurus est une forme de système de gestion des connaissances qui joint les propriétés d'un glossaire avec celles d'un thésaurus. Par conséquent, pour gérer cet ensemble de gestion des données terminologiques, nous combinons les fonctions assignées aux glossaires et celles employées pour gérer les thésaurus.

A travers cette fonctionnalité, un utilisateur pourra gérer en plus des relations qui existent entre les termes, les définitions de chacun d'eux. Aussi lui sera-t-il possible de consulter les termes qui sont associés à un glossaurus. Il pourra, en outre, détailler un terme en affichant ses relations et les définitions des termes qui lui sont liées suivant les types de relations.

5.4.1.4 Exporter des ensembles terminologiques basés sur SKOS

L'export des ensembles terminologiques basés sur SKOS est un des besoins fonctionnels de haut niveau de l'outil. En effet, comme le titre de la fonction le suggère, cette fonctionnalité permet de porter les ensembles de données terminologiques vers d'autres applications et ce, en créant un fichier XML. Ce dernier contient toutes les données d'un ensemble terminologique normalisé en SKOS ainsi que le mécanisme de mapping utilisé pour faire correspondre les propriétés de SKOS avec celles d'un système d'organisation des connaissances. La figure suivante affiche le contenu du thésaurus « Law Congress » après l'avoir exporté.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:skos="http://www.w3.org/2004/02/skos/core#"
- <skos:ConceptScheme rdf:about="LawCongress">
- <skos:Concept rdf:about="LawCongress.Loi_1975">
- <skos:inScheme rdf:resource="LawCongress" />
- <skos:prefLabel xml:lang="fr">Loi 1975 (handicap)</skos:prefLabel>
- <skos:narrower rdf:resource="LawCongress.Loi_1987" />
- </skos:Concept>
- <skos:Concept rdf:about="LawCongress.Loi_1987">
- <skos:prefLabel xml:lang="fr">Loi 1987 (emploi des personnes handicapées)</skos:prefLabel>
- <skos:inScheme rdf:resource="LawCongress" />
- <skos:broader rdf:resource="LawCongress.Loi_1975" />
- <skos:broader rdf:resource="LawCongress.Loi_2005" />
- </skos:Concept>
- <skos:Concept rdf:about="LawCongress.Loi_1988">
- <skos:prefLabel xml:lang="fr">Loi 1988 (recherche biomédicale)</skos:prefLabel>
- <skos:inScheme rdf:resource="LawCongress" />
- </skos:Concept>
- <skos:Concept rdf:about="LawCongress.Loi_2005">
- <skos:prefLabel xml:lang="fr">Loi 2005 (handicap)</skos:prefLabel>
- <skos:inScheme rdf:resource="LawCongress" />
- <skos:narrower rdf:resource="LawCongress.Loi_1987" />
- </skos:Concept>
- <skos:Concept rdf:about="LawCongress.Loi_2005_bis">
- <skos:prefLabel xml:lang="fr">Loi 2005 (droits des malades)</skos:prefLabel>
- <skos:inScheme rdf:resource="LawCongress" />
- </skos:Concept>
- </skos:ConceptScheme>
- </rdf:RDF>
```

Figure 5.13 - Contenu du thésaurus « Law Congress » après exportation

5.4.1.5 Élaboration de KOS

A travers cette fonctionnalité, l'utilisateur de l'outil de gestion des ensembles de données terminologiques basées sur SKOS sera en mesure de créer de nouveaux systèmes de gestion des connaissances conformes au standard SKOS.

L'utilisation de cette fonctionnalité n'est offerte qu'aux administrateurs qui possèdent les connaissances suffisantes relatives à la norme SKOS et qui sont aptes à établir des

correspondances entre les propriétés de SKOS et celles du nouvel ensemble terminologique à élaborer.

L'idée est d'offrir à l'administrateur le choix de créer soit un schéma de concepts, soit une collection et d'attribuer un nom significatif tel que thésaurus, « subject heading », ou une ontologie, et de les stocker dans les tables correspondantes dans le modèle conceptuel de SKOS.

Suivant ce choix, l'outil propose à l'administrateur une suite de questions pour déterminer quelles sont les propriétés SKOS à utiliser dans cet ensemble ainsi que la terminologie utilisée pour désigner la propriété. Naturellement, les relations ne peuvent être créées que s'il s'agit d'un schéma de concepts.

Ces propriétés une fois identifiées seront stockées dans les trois tables métiers servant de méta-classes.

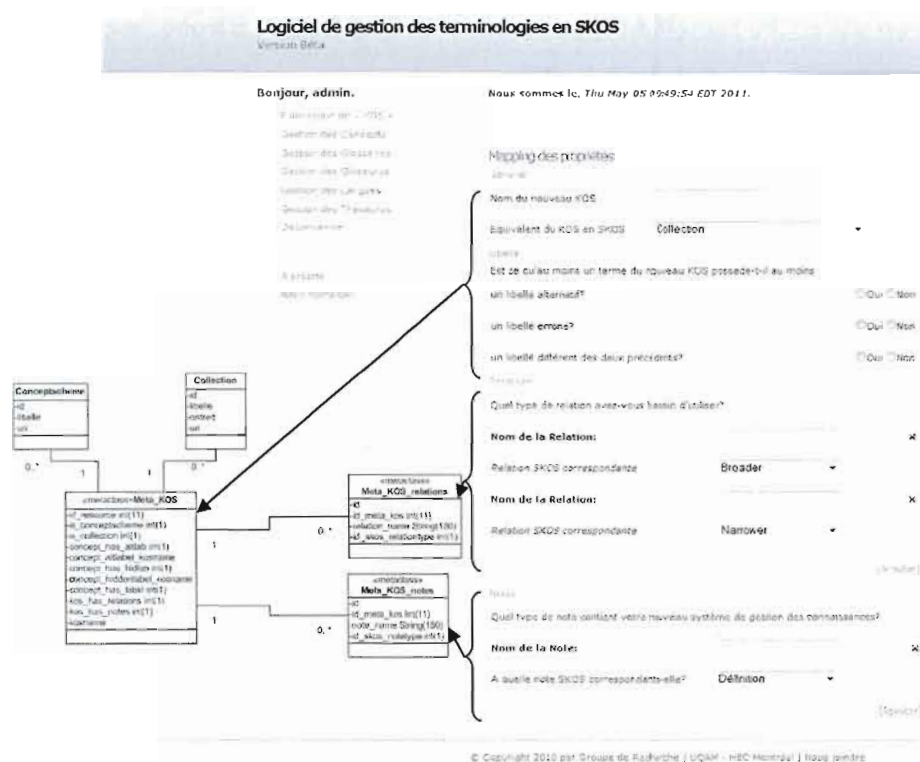


Figure 5.14 - Création d'un nouveau système de gestion des connaissances

5.4.1.6 Gestion des concepts SKOS

Ce module permet de gérer tous les concepts en SKOS et est destiné aux administrateurs de l'application qui ont une connaissance de SKOS. L'intérêt de cette fonctionnalité est de permettre à ses utilisateurs d'accéder directement aux concepts et à leurs propriétés en SKOS sans pour autant tenir compte de la nature de l'ensemble de données terminologiques auquel ils sont associés.

L'administrateur, à travers cette fonctionnalité, sera en mesure d'ajouter des nouveaux concepts SKOS, de les modifier, de les supprimer et de les associer à un ou plusieurs ensembles de données existants.

5.4.1.7 Gestion des langues

Cette fonctionnalité permet à l'administrateur de spécifier les différentes langues qui seront utilisées dans l'outil de gestion des données terminologiques. En effet, si nous prenons l'exemple de la gestion d'un glossaire, un simple utilisateur peut effectuer diverses opérations. Une des plus simples concerne la création d'une nouvelle définition. Cette définition consistera en une chaîne de caractères qui sera écrite dans une des langues qui a été spécifiée par l'administrateur.

Donc, le module de gestion des langues a pour but de définir les langues qui seront utilisées pour la réalisation de certaines opérations telles que l'ajout et la modification. Il est à signaler que la langue par défaut est le français qui ne peut être supprimé.

5.5. MECANISME DE « MAPPING » ENTRE LES KOS ET SKOS

Une fois les besoins et spécifications du logiciel établies, nous développons notre outil de gestion des données terminologiques qui viendra se greffer à l'« engin SKOS ».

La principale difficulté, relative au développement de cet outil, concerne les règles de correspondance entre le modèle conceptuel de SKOS et les différentes propriétés d'un ensemble de données terminologiques. En effet, l'outil que nous voulons construire doit être simple d'utilisation et intuitif. Il doit permettre à n'importe quel utilisateur de le manipuler sans avoir connaissance de SKOS et de ses propriétés. Nous devons, par conséquent, proposer un outil qui offre visuellement les mêmes propriétés qu'un KOS

quelconque, comme par exemple un glossaire, un thésaurus, mais qui puisse posséder, en arrière-plan, un mécanisme permettant de faire correspondre les propriétés de celui-ci avec celles du modèle conceptuel de SKOS. Étant donné que notre outil propose par défaut de gérer des glossaires, des thésaurus et des glossaurus qui est une forme combinée de glossaires et de thésaurus, nous nous arrêterons à mapper les propriétés des deux premières sortes de KOS.

5.5.1 Règles de correspondance pour les glossaires

Un glossaire est un type de système d'organisation des données. Les éléments représentatifs d'un tel ensemble sont :

- L'intitulé du glossaire.
- Des termes.
- Des libellés.
- Des définitions.
- La langue.

En effet, un glossaire possède un nom permettant de facilement l'identifier. Généralement, ce nom est représentatif d'un domaine particulier telles que la biologie, l'aviation, etc. Par ailleurs, un glossaire est constitué d'un ensemble de termes représentés par leur libellé. Ces derniers représentent la terminologie utilisée dans une spécialité pour communiquer, par exemple, un processus ou un événement en utilisant des mots techniques. De plus, un glossaire contient un ensemble de définitions relatives à des termes qui sont représentés, tout comme leurs termes, dans une ou plusieurs langues. Cette définition permet de donner un sens au terme pour ainsi faciliter son utilisation.

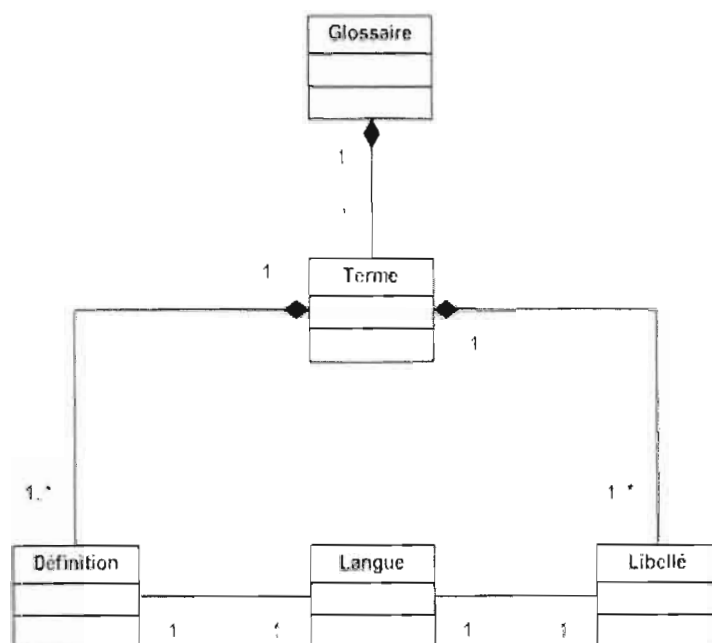


Figure 5.15 - Représentation UML d'un glossaire

5.5.1.2 Le glossaire sous SKOS

Étant donné qu'un glossaire est constitué d'un ensemble de termes (Wright Sue Ellen et Budin Gerhard, 1997), nous pouvons le représenter, en SKOS, sous deux formes. SKOS permet de représenter les divers ensembles soit en utilisant les schémas de concepts soit en recourant aux collections.

Pour réaliser notre choix, nous nous sommes reposés sur la définition donnée aux glossaires. Selon Wright Sue Ellen et Budin Gerhard (1997), un glossaire contient des termes **ordonnés** alphabétiquement. Donc, le choix de l'une ou de l'autre propriété SKOS dépendra essentiellement de leur capacité à offrir ou pas la possibilité de classer les concepts. Or, un schéma de concept ne permet pas d'ordonner les concepts qu'il contient. Il permet, contrairement aux collections, de hiérarchiser des concepts et non de les classer. Ainsi à travers les définitions proposées par Wright Sue EllenBudin Gerhard (1997) et celles données aux collections ainsi que les schémas de concepts en SKOS, nous sommes arrivés à établir une première règle de correspondance.

Notion	Propriété SKOS correspondante
Glossaire	Skos : collection

Tableau 5.1 - Règle de correspondance pour un glossaire

5.5.1.3 Le terme d'un glossaire en SKOS

Un terme est l'élément principal constituant un glossaire. Il est défini par le triplet (libellé, définition, langue). Dans le méta-modèle SKOS, la notion de **terme** n'existe pas. En effet, rappelons-le, SKOS repose **uniquement sur des concepts**. Nous pourrions poser comme hypothèse qu'un terme correspond à un concept dont le libellé correspond à celui du terme. Cependant, un concept SKOS est une notion plus large et plus abstraite que la notion de terme. Nous voilà alors confronté à une première difficulté, celle de faire correspondre la notion de **terme** à celle d'**une propriété de SKOS**.

Afin de résoudre ce problème, nous nous appuyons sur la recommandation faite par le W3C. Celle-ci définit les concepts comme étant **indépendants** des mots utilisés pour les étiqueter. Elle définit, par ailleurs, que chaque entité introduite ne peut être représentée que sous la forme d'une ressource, qui pourrait être soit un schéma de concepts, soit une collection, soit une note, soit un concept.

Partant de ces recommandations, nous pouvons considérer, dès lors, qu'un terme est l'entité à introduire dans le méta-modèle SKOS et qui possède comme libellé le mot utilisé pour l'étiqueter. Dans le cas d'un terme, il ne pourra être représenté que sous la forme d'un concept, vues les définitions données aux autres ressources par le W3C.

Pour illustrer ce choix, nous pouvons prendre l'exemple d'un terme du glossaire « Génomique ». Si l'on considère le terme « gène », nous pourrions le représenter comme un concept sans intitulé possédant un URI et ayant pour étiquette le mot « gène » qui sera associé à un des trois labels SKOS.

Les labels SKOS permettent, comme le nom le suggère, d'étiqueter les concepts. Comme nous l'avons déjà vu, il existe trois sortes de label SKOS.

- PrefLabel : permet d'assigner à un concept un label préféré ; c'est le label qui décrit au mieux le concept auquel il est rattaché.
- AltLabel permet d'affiner la description d'un concept SKOS.
- HiddenLabel sert à prévenir les fautes d'orthographe lors d'une saisie de recherche.

L'étude préalable du méta-modèle SKOS nous laisse décider que le label préféré SKOS « skos :prefLabel » est le plus adapté.

En effet, si on regarde de près les recommandations du W3C, nous pouvons observer que chacun des labels SKOS a une utilisation spécifique. Le « hidden label » et le label alternatif sont respectivement employés pour contenir les mots erronés et les mots dont l'utilisation est moins courante. Or, tout terme possède un libellé d'usage courant, connu de tous.

Par conséquent, il est impossible d'employer les deux propriétés « skos :hiddenLabel » et « skos :altLabel » pour représenter le libellé d'un terme d'un glossaire. Le libellé préféré « skos :prefLabel » sera, donc, la propriété à contenir le libellé d'un terme.

Nous pouvons, dès lors, ériger les règles suivantes :

Notions	Propriétés SKOS correspondantes
Terme	Skos :Concept
Libellé d'un terme	Skos :PrefLabel

Tableau 5.2 - Règles de correspondance pour un terme et son libellé

5.5.1.4 La définition en SKOS

Le glossaire, en plus d'un ensemble de termes qu'il contient, comprend aussi un ensemble de définitions permettant de détailler ses termes. Le méta-modèle SKOS propose déjà dans son modèle une propriété permettant de représenter les définitions. Les notes SKOS

sont employées à ces fins. En effet, « skos :note » est une entité générique qui comporte plusieurs sortes de notes dont des historiques d'une ressource, des exemples et surtout des définitions.

Ainsi, dans le cas d'une définition d'un glossaire, nous utiliserons la « skos :definition », une spécificité de « skos :note », comme règle de correspondance entre le modèle conceptuel d'un glossaire et celui de SKOS pour représenter l'élément « définition ».

Notion	Propriété SKOS correspondante
Définition	Skos :defintion

Tableau 5.3 - Règle de correspondance pour une définition d'un glossaire

5.5.1.5 La langue sous SKOS

Concernant la langue, SKOS prévoit l'utilisation d'annotation. Celle-ci permet d'ajouter une précision quant à la langue utilisée pour écrire la chaîne de caractères représentative du libellé d'un terme ou d'un glossaire ou autre. Cette annotation est matérialisée par le symbole « @ » après chaque chaîne de caractères.

Notion	Propriétés SKOS correspondante
Langue	@langue

Tableau 5.4 - Règle de correspondance pour la langue

5.5.1.6 Conclusion

Pour résumer, nous illustrons les règles de correspondances entre les éléments d'un glossaire et les propriétés SKOS dans le tableau suivant :

Notions	Propriétés SKOS correspondantes
Glossaire	Skos : collection
Terme	Skos :Concept
Libellé d'un terme	Skos :PrefLabel
Définition	Skos :defintion
Langue	@langue

Tableau 5.5 - Résumé des règles de correspondance

5.5.2 Règles de correspondance pour les thésaurus

Tout comme les glossaires, les thésaurus contiennent des termes. Cependant ces derniers sont hiérarchisés grâce à différents types de relations.

Comme vous avez pu l'observer, un thésaurus n'emploie pas de concepts. Il contient des termes qui sont identiques à ceux contenus dans un glossaire. Ces termes sont reliés entre eux grâce à des relations hiérarchiques, synonymiques et associatives. Ce sont les éléments de base représentatifs d'un thésaurus. Il est donc nécessaire de trouver les éléments analogues dans SKOS permettant de les représenter afin d'aboutir à des règles de correspondance.

Il est à noter qu'un thésaurus dispose aussi d'un ensemble de termes descripteurs et non descripteurs. Ils servent respectivement à l'indexation et à la recherche. Ces propriétés sont importantes pour les utilisateurs des thésaurus. Il est, par conséquent, impératif de trouver les **propriétés correspondantes** dans le méta-modèle SKOS et de pouvoir distinguer ces deux aspects.

Pour procéder à l'identification des différentes règles de correspondance entre les propriétés d'un thésaurus et les éléments de SKOS, nous commençons par identifier les

éléments constitutifs d'un thésaurus. Généralement, nous observons les propriétés suivantes :

- Le thésaurus
- Les termes
- Les libellés
- Les relations

que nous avons schématisées dans la figure suivante :

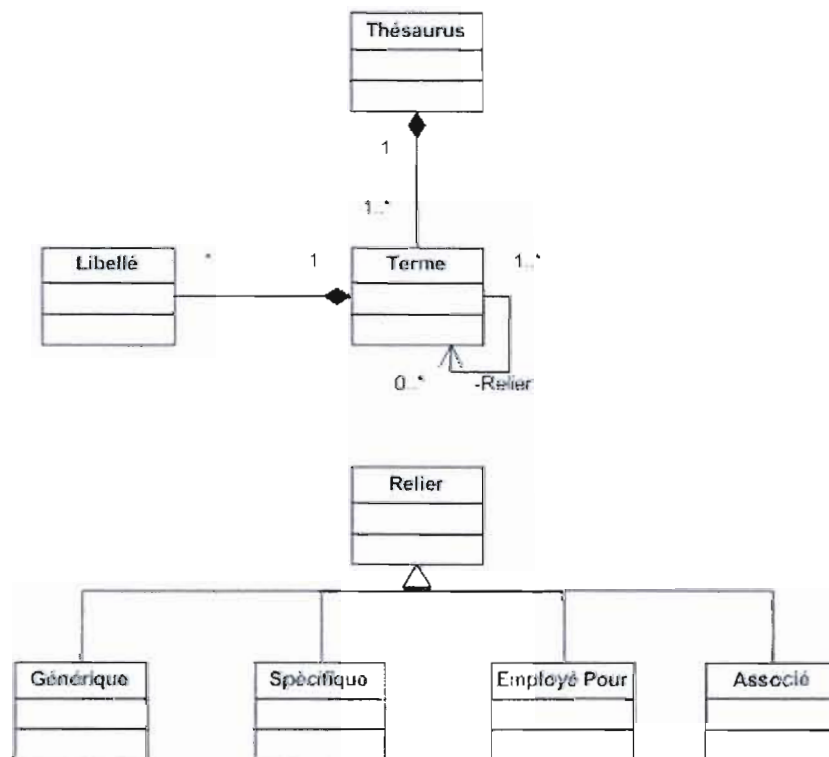


Figure 5.16 - Modèle conceptuel des thésaurus

5.5.2.2 Thésaurus sous SKOS

Selon, l'organisation internationale de la standardisation, un thésaurus est un « vocabulaire d'un langage d'indexation contrôlé organisé formellement de façon à

expliciter les relations à priori entre les notions » (ISO5964, 1985). Nous pouvons alors déduire qu'un thésaurus est constitué d'un **ensemble de notions**, entre autre les termes, **organisées grâce à des relations**.

A partir de cette définition, nous pouvons déjà tenter de trouver la propriété correspondante en SKOS qui pourrait représenter un ensemble de termes et qui permettrait de les organiser.

Le document de référence de SKOS stipule que les schémas de concepts permettent de hiérarchiser des concepts grâce aux relations sémantiques de SKOS. Cette propriété répond aux contraintes imposées par ISO dans sa définition des thésaurus.

Nous pouvons alors conclure aux deux règles de correspondances suivantes :

Notion	Propriété SKOS correspondante
Thésaurus	Skos : conceptscheme
Relation	Skos : semanticRelation

Tableau 5.6 - Règle de correspondance d'un thésaurus et des relations en SKOS

5.5.2.3 Terme d'un thésaurus en SKOS

Les termes d'un thésaurus sont identiques aux termes d'un glossaire. Tous les deux possèdent les mêmes caractéristiques. Ainsi, pour établir les règles de correspondance, nous pouvons utiliser les mêmes que celles établies pour les termes d'un glossaire, soit les règles représentées dans le tableau suivant :

Notions	Propriétés SKOS correspondantes
Terme	Skos : Concept

Libellé d'un terme	Skos :PrefLabel
--------------------	-----------------

Tableau 5.7 - Règles de correspondance pour un terme et son libellé

5.5.2.4 Les relations d'un thésaurus en SKOS

Nous avons pu identifier grâce aux définitions d'ISO et du W3C, concernant respectivement les thésaurus et les concepts schèmes, que les relations entre les termes d'un thésaurus équivalaient aux relations sémantiques de SKOS. Cependant, la correspondance que nous avons réalisée est généraliste. Les deux types de relations comprennent un sous ensemble de variétés de relations.

Pour un thésaurus, il existe essentiellement trois sortes de relations :

- *Hiérarchiques*, c'est-à-dire, entre un terme générique et un autre spécifique. Cette relation comprend deux sortes de hiérarchies : spécifique et générique.
- *Associatives* appliquées uniquement aux termes descripteurs afin de les enrichir sémantiquement en établissant des liens avec des termes d'un domaine connexe.
- *Équivalences* établies entre un terme descripteur et un terme non descripteur, entre des termes qui sont synonymes ou quasi-synonymes.

Comme nous l'avons noté plus haut, le méta-modèle SKOS propose aussi une catégorie de relations, « skos :semanticRelation », permettant de lier les concepts entre eux et qui à son tour propose six sortes de relations que nous avons réduit à quatre.

En effet, deux des propriétés qui composent les relations sémantiques de SKOS, « skos :broaderTransitive » et « skos :narrowerTransitive », sont utilisées non pas pour certifier que des liens hiérarchiques existent entre les concepts mais pour les déduire et permettre ainsi un accès direct ou indirect entre eux.

Ainsi, les quatre propriétés SKOS conservées sont les suivantes :

- Skos :narrower
- Skos :broader
- Skos :related
- Skos :closeMatch

La définition énoncée par le W3C explique clairement le rôle de chacun des deux premiers types de relations. Elle énonce « **A hierarchical link between two concepts indicates that one is in some way more general ("broader") than the other ("narrower")** » (A. Miles et al, 2009).

En d'autres termes, les relations hiérarchiques sont composées de:

- relations génériques permettant de relier un terme générique à un terme spécifique.
- relations spécifiques fonctionnant à l'inverse des relations hiérarchiques.

Dans le méta-modèle SKOS, deux propriétés permettent de représenter respectivement ces deux liens : Les « skos:broader » et les « skos:narrower ».

Le W3C explique davantage « **An associative link between two concepts indicates that the two are inherently "related", but that one is not in any way more general than the other.** » (A. Miles et al, 2009). Or, la propriété « skos:related » est la seule à être conforme à cette description.

Nous pouvons alors à partir de cette définition déduire les trois règles de correspondances suivantes :

Notions	Propriétés SKOS correspondantes
Relation générique	Skos :broader

Relation spécifique	Skos :narrower
Relation associative	Skos :related

Tableau 5.8 - Règles de correspondance entre les relations d'un thésaurus et SKOS

Concernant le dernier type de relation, la relation d'équivalence, le méta-modèle SKOS propose deux propriétés qui pourraient la représenter. En effet, si on se fie à A.Miles et al., il existe « skos :exactMatch » et « skos :closeMatch » qui pourraient la représenter. La seule différence entre ces deux propriétés réside dans le fait que l'une est transitive, tandis que l'autre ne l'est pas.

Donc, pour pouvoir faire notre choix, il faudrait d'abord savoir que si le fait qu'un terme X d'un thésaurus est un terme équivalent à Y et que Y est un terme équivalent à Z, alors ceci induit directement à la conclusion que X est un terme équivalent à Z.

Par conséquent, nous devrions avant tout choix nous poser la question sur la transitivité d'une relation d'équivalence dans un thésaurus.

La norme ISO, qui gère les principes de construction d'un thésaurus, ne discute pas clairement de la transitivité d'une relation d'équivalence. Toutefois, elle la présente comme étant applicable à deux sortes de termes :

- *Termes synonymes* qui sont des termes « dont on peut considérer la signification comme **identique** dans une large gamme de contextes, de sorte qu'ils sont **virtuellement interchangeables**. »(ISO 2788, 1986).
- *Termes quasi-synonymes* qui sont des termes « dont la **signification est généralement considérée comme différente** dans l'usage ordinaire, mais qui sont traités comme des synonymes » (ISO 2788, 1986).

D'après ces deux définitions, nous pouvons aboutir aux deux postulats suivants :

- Si la relation d'équivalence A lie deux termes synonymes X et Y et que la relation B lie les deux termes synonymes Y et Z, alors nous pouvons

conclure par transitivité qu'il existe une relation implicite C entre X et Z. Dans ce cas, nous parlons de transitivité de la relation et en conséquence, nous utiliserons la propriété « **skos :exactMatch** » pour la représenter en SKOS car elle permet, en outre, cette transitivité.

- Si la relation d'équivalence A lie deux termes quasi-synonymes X et Y et que la relation B lie les deux termes synonymes ou quasi-synonymes Y et Z, alors aucune relation ne peut être déduite entre X et Z. Dès lors, la transitivité de la relation n'existe pas et la propriété « **skos :closeMatch** » sera la propriété appropriée pour représenter cette relation en SKOS.

Ces deux postulats nous permettent d'ériger les règles de correspondance suivantes :

Notions	Propriétés SKOS correspondantes
Relation d'équivalence entre termes quasi-synonymes	Skos :closeMatch
Relation d'équivalence entre termes synonymes	Skos :exactMatch

Tableau 5.9 - Règles de correspondance entre les relations synonymiques d'un thésaurus et SKOS

5.5.2.5 Conclusion

Après avoir établi les différentes règles de correspondances, nous résumons l'ensemble dans le tableau suivant :

Notion	Propriété SKOS correspondante
Thésaurus	Skos : conceptscheme

Relation	Skos :semanticRelation
Terme	Skos :Concept
Libellé d'un terme	Skos :PrefLabel
Relation hiérarchique	
Relation générique	Skos :broader
Relation spécifique	Skos :narrower
Relation associative	
Relation associative	Skos :related
Relation d'équivalence	
Relation d'équivalence entre termes quasi-synonymes	Skos :closeMatch
Relation d'équivalence entre termes synonymes	Skos :exactMatch

Tableau 5.10 - Résumé des règles de correspondances

5.6. CONCLUSION

L'outil que nous avons développé est complet. Les différentes fonctionnalités sont opérationnelles et le mécanisme de correspondance est fiable. En effet, grâce à l'identification des différents éléments de SKOS et à leur mise en correspondance avec

les divers éléments des ensembles terminologiques, il est désormais plus facile pour un simple utilisateur de les transporter d'une application à une autre. Le fichier produit lors d'une exportation d'un KOS contient, en plus de leurs données normalisées, les règles qui permettent à l'importateur de les intégrer dans une autre solution. Ainsi, l'outil permettrait une meilleure interopérabilité.

Par ailleurs, cet outil, grâce à ces différentes fonctionnalités, offre une bonne gestion des systèmes de gestion des connaissances. Avec les différentes opérations dont il dispose, un utilisateur pourrait manipuler les ensembles de données terminologiques basés sur SKOS d'une manière intuitive et simple.

Théoriquement, l'outil devrait être fiable étant donnée la méthode que nous avons suivie pour sa réalisation. Cependant, sur le plan pratique, il faudrait effectuer des tests auprès d'utilisateurs, personnes ou organismes, qui recourent régulièrement aux systèmes d'organisation des connaissances afin d'évaluer notre outil. Le chapitre suivant sera consacré en partie à ce dernier point.

CHAPITRE VI

EXPÉRIMENTATION

6.1. INTRODUCTION

Ce chapitre a pour but, d'une part, de démontrer la capacité de notre outil à gérer et à manipuler des ensembles de données terminologiques basés sur SKOS et d'autre part, de valider l'outil en soi. Pour ce faire, nous procédons au cours de ce chapitre, à des expérimentations sur des glossaires et des thésaurus existants détenus par deux organismes distincts.

Il est à rappeler que notre outil offre par défaut la gestion des deux types de KOS cité plus haut. Ainsi, nous nous intéressons plus à ces deux ensembles qu'au reste des KOS recensés dans le chapitre 2.

6.2. CREATION ET MANIPULATION D'UN GLOSSAIRE

Afin de tester la fonctionnalité de gestion des glossaires, nous insérons à travers cette étude de cas, un glossaire élaboré par la Banque du Canada.

6.2.1 Présentation du cas d'étude et du scénario de test

La Banque du Canada est « la banque centrale du pays. Elle n'est pas une institution financière et elle n'offre pas de services bancaires au public. Elle est plutôt responsable de la politique monétaire, des billets de banque, du système financier, de la gestion financière du Canada » (Banque du Canada, 2010).

Une des activités, que nous pouvons citer à titre d'exemple, concerne leur processus de prise de décision. La description de ce processus contient des termes qui sont plus ou moins compréhensibles.

Afin de permettre une meilleure compréhension, la Banque du Canada dispose d'un glossaire bilingue. Il contient différents termes techniques, économiques et financiers ainsi que leur définition dans deux langues : le français et l'anglais. La figure suivante présente une partie du glossaire de la banque.

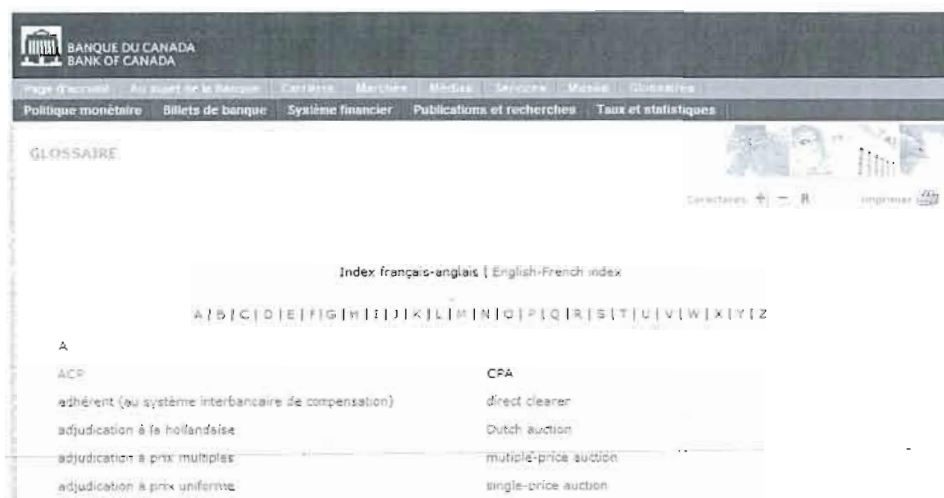


Figure 6.1 - Glossaire de la Banque du Canada (source : <http://www.bank-banque-canada.ca/fr/glossaire/glossaire.html>)

6.2.2 Expérimentation

6.2.2.1 Scénario de l'expérimentation

Afin de nous assurer que notre outil permet de manipuler efficacement les glossaires, nous procédons à partir du glossaire de la Banque du Canada à :

- La création d'un glossaire à travers notre outil
- L'insertion d'une partie des termes qu'il contient
- Vérification des données dans la BD grâce à des scripts
- Edition de certaines données à travers l'outil

- Suppression du glossaire
- Vérification du contenu de la base de données en recourant à des scripts
- Analyse et synthèse des résultats.

6.2.2.2 Exécution du scénario

Pour réaliser notre expérimentation, nous créons, tout d'abord, un glossaire en l'intitulant « Banque du Canada » et désignant son URI par « banqueducanada ». Suite à cette opération, l'outil nous retourne vers l'interface de gestion des glossaires où nous voyons apparaître ce nouveau glossaire.



Figure 6.2 - Processus de création du glossaire « Banque du Canada »

Nous procédons ensuite à l'insertion des différents termes contenus dans le glossaire recueilli de la Banque. Pour chaque terme, nous introduisons un libellé préféré et une définition dans chacune des deux langues ainsi que des exemples au cas où il en posséderait.

Une fois cette opération achevée, l'outil procède à l'affichage des différents éléments introduits du terme ainsi que les fonctionnalités que l'utilisateur pourrait utiliser pour effectuer d'autres types d'opérations. La figure qui suit, présente le processus d'insertion d'un nouveau terme « assouplissement quantitatif » ainsi que la manière dont celui-ci est affiché après son insertion.

Glossaire des Termes de Banque du Canada

Nom du concept: assouplissement

URI: anqueducanada.assouplissement

Spécifications:

Nom du terme: assouplissement quantitatif

Acronymes:

Notes:

Définition:

échéance rapprochée pourrait aussi contribuer à renforcer l'incidence de déclarations conditionnelles au sujet de la trajectoire future des taux d'intérêt.

Exemple:

Langue: Français

Sauvegarder Annuler

Glossaire des Termes de Banque du Canada

Cherchez: [Rechercher]

Nom(s) du Terme	Langue(s)	URI
assouplissement	Français(FR)	anqueducanada.assouplissement
quantitatif	Français(FR)	anqueducanada.assouplissement_quantitatif

URI: anqueducanada.assouplissement_quantitatif

Notes:

Définition:

Langue: Français(FR)

Achat ferme par la banque centrale d'actifs financiers financé par la création de zolces de règlement excédentaires (c'est-à-dire les réserves que détiennent les institutions financières à la banque centrale). De tels achats feraient augmenter le prix des actifs acquis (qui pourraient être aussi bien des titres d'État que des actifs du secteur privé) et stimuler leur rendement. La hausse des soldes de règlement mis à la disposition des participants directs au système de transfert de paiements de grande valeur encouragerait ceux-ci à acheter des actifs ou à accroître l'offre de crédit aux ménages et aux entreprises. Il en résulterait un accroissement de l'offre de dépôts (ou des agrégats monétaires) et une hausse additionnelle de la demande d'autres actifs financiers, ce qui tiendrait leurs prix vers le haut et leurs rendements vers le bas. L'achat par la banque centrale de bons du Trésor et d'obligations d'État assorties d'une échéance rapprochée pourrait aussi contribuer à renforcer l'incidence de déclarations conditionnelles au sujet de la trajectoire future des taux d'intérêt. [1234]

Langue: Anglais(EN)

The outright purchase by a central bank of private sector assets in certain credit markets that are important to the functioning of the financial system but that are temporarily impaired. NOTE: The objective of credit easing is to reduce risk premiums and improve liquidity and trading activity in these markets. This would, in turn, stimulate credit flows and aggregate demand. Credit easing does not need to be financed through an expansion of settlement balances. It could, instead, be financed either by reducing holdings of other assets or by increasing government deposit liabilities, so that the monetary base remains unchanged. In other words, the impact of credit easing on the level of settlement balances could be "sterilized." [1234]

Aucune Exemple n'est associée à ce terme de votre glossaire.

Figure 6.3 – Processus d'insertion d'un nouveau terme dans le glossaire de « Banque du Canada »

En observant cette figure, nous remarquons que l'utilisateur a une vue simpliste des informations introduites. Elles sont présentées tel dans un glossaire traditionnel. Il ne se douterait pas que le contenu a été normalisé selon le standard SKOS. D'ailleurs, nous non plus nous ne pourrions pas faire cette distinction hormis de passer par l'analyse du contenu de la base de données.

Ainsi, pour s'assurer que les informations ont été stockées aux bons endroits, nous analysons le contenu de la base de données et nous vérifions au travers des tableaux de correspondance que les informations créées correspondent bien aux mêmes éléments de SKOS mappés précédemment. Pour ce faire, pour chaque règle de correspondance d'un mapping d'un glossaire, nous bâtissons les résultats attendus théoriquement et nous vérifions par la suite, grâce à des scripts, s'ils correspondent bien au contenu de la base de données de notre outil.

Pour accomplir cette tâche, nous développons un script basé sur SQL qui permet de recueillir et d'afficher le contenu des différents éléments qui devraient à priori être affectés par l'insertion d'un nouveau terme. Ce script requiert comme paramètres l'identifiant du terme ainsi que celui du système de gestion des connaissances qui le contient. Le script implémenté est dès lors comme suit:

```
public String executeScript()
{
    String SQL = "SELECT concept.libelle as libconcept, concept.source as conceptsource, collection.libelle as collectionlibelle, "
        + "collection.prefix as collectionprefix, collection.suffix as collectionsuffix "
        + "FROM concept, collection, member "
        + "WHERE concept.id = '"+this.idConcept+"' AND collection.id = '"+idMsc+"' AND member.idconcept=concept.id "
        + "AND member.idcollection = collection.id";

    String SQL2 = "SELECT labelsks.idlangue as labelsksid, labelsks.idlabeltype as idlabeltype, labelsks.libelle as labelskslibelle "
        + "FROM labelsks, labeltype "
        + "WHERE labelsks.idconcept = '"+this.idConcept+"' AND labelsks.idlabeltype = labeltype.id AND "
        + "labeltype.libelle = 'prefLabel'";

    String SQL3 = "SELECT note.notestring as notestring, note.idnotetype as idnotetype, note.idlangue as idlanguenote, "
        + "langue.abbreviation as langabbrv, langue.id as idlangue "
        + "FROM note, notetype, langue "
        + "WHERE note.idconceptsource = '"+this.idConcept+"' AND note.idnotetype = notetype.id AND note.idlangue = langue.id "
        + "AND notetype.libelle = 'Definition'";
}
```

Figure 6.4 - Une partie de la fonction d'exécution du script sur un glossaire

Après avoir exécuté ce script, nous obtenons, à l'affichage, les résultats présentés dans la figure qui suit. Il est à noter que les chaînes de caractères « stringNote » ont été délibérément réduits pour diminuer la taille de la figure et que l'identifiant du terme a été choisi aléatoirement.

PROCESSING SCRIPT...

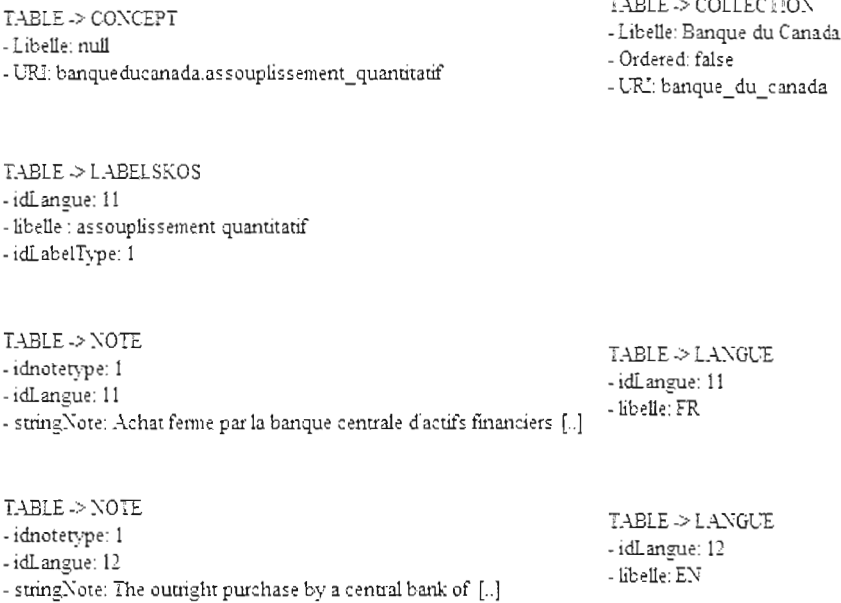


Figure 6.5 - Résultats de l'exécution du script d'expérimentation du mécanisme de « mapping » sur un glossaire

A partir de ces résultats nous réalisons notre analyse. Nous observons que les résultats produits théoriquement sont identiques à ceux stockés dans la BD de l'outil. Le tableau ci-dessous, résume l'analyse que nous avons réalisée pour l'exemple de la figure 5.3.

Notions	Propriétés SKOS correspondantes	Résultats théorique avec l'exemple de la figure 52	Résultats trouvés dans la BD
Glossaire	Skos : collection	Table ='collection' - Libelle = 'Banque du Canada' - Ordered= 'false' - URI = 'banque_du_canada'	✓

Terme	Skos :Concept	Table= 'concept' - Id= 'équivalent à une valeur X ' - Libelle = 'assouplissement_quantitatif' - URI= 'banqueducanada.assouplissement_qu antitatif'	✓
Libellé d'un terme	Skos :PrefLabel	Table= 'labelskos' - idconcept= 'X' - idLangue = '11' - libelle = 'assouplissement quantitatif' et - idconcept= 'X' - idLangue = '12' - libelle = 'quantitative easing' - idLabelType = 'valeur du id correspondant à « PrefLabel » dans la table labeltype'	✓
Définition	Skos :defintion	Table= 'note' - idnotetype= 'valeur correspondante au type « définition » dans la table notetype annoté dans cet exemple par N' - Idconceptsource= 'X ' - idLangue= '11' - stringNote = 'Achat ferme par la banque centrale d'actifs financiers [..]' et - idnotetype= 'N' - Idconceptsource= 'X ' - idLangue= '12' - stringNote = 'The outright purchase by a central bank of financial assets [.]'	✓
Langue	@langue	Table= 'langue' - id= '11' - libelle= 'Fr ' et - id= '12' libelle= 'En'	✓

Tableau 6.1 - Evaluation du contenu de la BD

Concernant les autres types d'opérations permettant la manipulation des données dans ce glossaire, nous avons réitéré le même travail fait lors d'une opération de création. Au sujet des opérations d'édition, nous avons réalisé les modifications possibles de quelques termes dont celui présenté ci lors d'une création. Pour ce faire, nous avons réutilisé le

même script cité plus haut pour obtenir les données à analyser et nous les avons, ensuite, comparées avec celles qui ont été établies théoriquement. Le résultat obtenu est comme, escompté et les modifications que nous avons effectuées ont été correctement stockées dans la base de données représentative du standard SKOS.

Par ailleurs, nous avons effectué des suppressions de termes du glossaire. Aucune trace des éléments supprimés n'a été trouvée. Ceci s'explique par le fait que l'outil recourt à la suppression en cascade qui permet d'effacer toutes les informations d'un objet identifié dans toutes ses relations.

6.2.3 Conclusion et limites

La manipulation d'un glossaire à partir de notre outil n'a pas été une tâche compliquée. Les différentes fonctionnalités qui permettent de gérer un glossaire fonctionnent correctement. En effet, les tests d'insertion, d'édition et de suppression, nous permettent d'affirmer que notre outil est apte à gérer efficacement des glossaires. Par ailleurs, les tests réalisés au niveau de la BD, nous ont montré que les informations introduites n'ont subi aucune perte. D'ailleurs, grâce au mécanisme de mapping, elles ont été stockées et normalisées sous SKOS.

6.3. CREATION ET MANIPULATION D'UN THESAURUS

La gestion des thésaurus est une des autres fonctionnalités implémentées par défaut dans le prototype que nous avons développé. Cette fonctionnalité comme décrite précédemment permet de gérer et de manipuler des thésaurus normalisés en SKOS. Afin de vérifier la capacité de cette fonctionnalité à gérer efficacement ce type d'ensemble de données terminologique, nous procédons à son expérimentation au travers d'un exemple de thésaurus économique du gouvernement du Québec.

6.3.1 Présentation du cas d'étude

Le thésaurus de l'activité gouvernementale a été élaboré par le gouvernement du Québec pour servir de référentiel aux personnes désirant faire « une recherche efficace dans les sites Web de l'Administration québécoise. »

Ce thésaurus contient « quelques milliers de termes couvrant l'ensemble des grands champs ou des domaines de l'activité gouvernementale (administration publique, agriculture, économie, éducation, environnement, finance, justice, santé, etc.) ».

Étant donné le nombre important des mots et expressions renfermés dans ce thésaurus, nous nous intéresserons à une partie de ce thésaurus, plus spécialement, celui couvrant le domaine de l'économie. La figure qui suit nous donne un aperçu sur ce thésaurus.

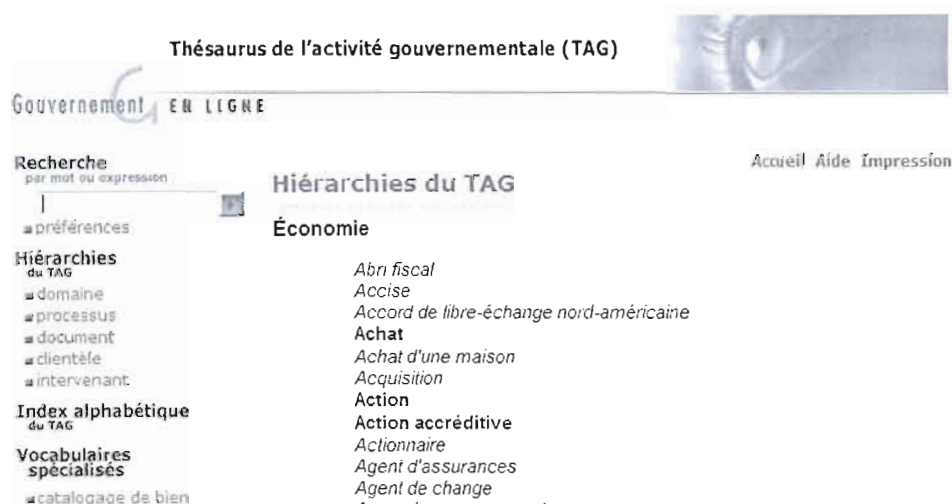


Tableau 6.2 - Thésaurus de l'Activité Gouvernementale (source : <http://www.thesaurus.gouv.qc.ca/tag/consultation/mt/tag.do?d=exp&id=3173#3173>)

A partir de ce thésaurus, nous procédons à l'application d'un scénario d'expérimentation. Ce dernier est explicité dans la section d'après.

6.3.2 Expérimentation

6.3.2.1 Scénario de l'expérimentation

Le scénario d'expérimentation de la fonctionnalité de gestion des thésaurus est identique à celui réalisé pour les glossaires. Celui-ci, suivantes :

- La création d'un thésaurus à partir de « Skos Tool »
- L'insertion d'une partie des termes qu'il contient
- Vérification des données dans la BD grâce à des scripts

- Edition de certaines données à travers l'outil
- Suppression du thésaurus
- Vérification du contenu de la base de données en recourant à des scripts
- Analyse et synthèse des résultats.

6.3.2.2 Exécution du scénario

Pour procéder à la réalisation de notre scénario d'expérimentation, nous commençons par créer notre thésaurus en l'intitulant « Economie ». La figure qui suit présente les interfaces et le processus de création d'un thésaurus.

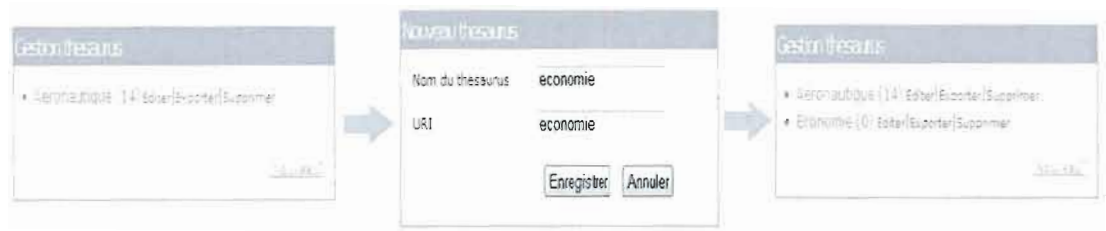


Figure 6.6 - Processus de création d'un thésaurus

Une fois le thésaurus créé, nous passons à l'exécution de la seconde étape de notre scénario. Pour ce faire, nous commençons tout d'abord à introduire une partie des termes économiques du thésaurus. Ensuite, nous procédons à l'insertion des relations existantes entre eux en éditant leurs relations. Certes la manière de procéder prend du temps néanmoins, elle nous évite de tomber sur le cas où un terme faisant partie d'une relation, n'existe pas. Pour illustrer cette opération d'ajout, la figure ci-dessous, nous donne un aperçu de l'interface que nous avons utilisée pour insérer le terme « Crédit ».

Figure 6.7 - Insertion du terme « Crédit »

Comme nous pouvons l'apercevoir, il est possible d'ajouter des relations à un terme à partir de deux interfaces différentes dépendamment de la stratégie adaptée par l'utilisateur. Étant donné que nous avons employé une stratégie d'ajout des termes dans une première étape et d'insertion des relations en une seconde, nous procéderons alors à l'ajout des relations relatives au terme « crédit » à partir de la seconde interface, c'est-à-dire, celle qui est à droite de la figure ci-dessus.

Pour spécifier de nouvelles relations au terme « crédit », nous consultons d'abord le thésaurus gouvernemental et nous cherchons ce terme pour détailler ses relations afin de les ajouter dans notre application.

Les différentes relations insérées, nous obtenons après affichage détaillé du terme « crédit » la figure qui suit.

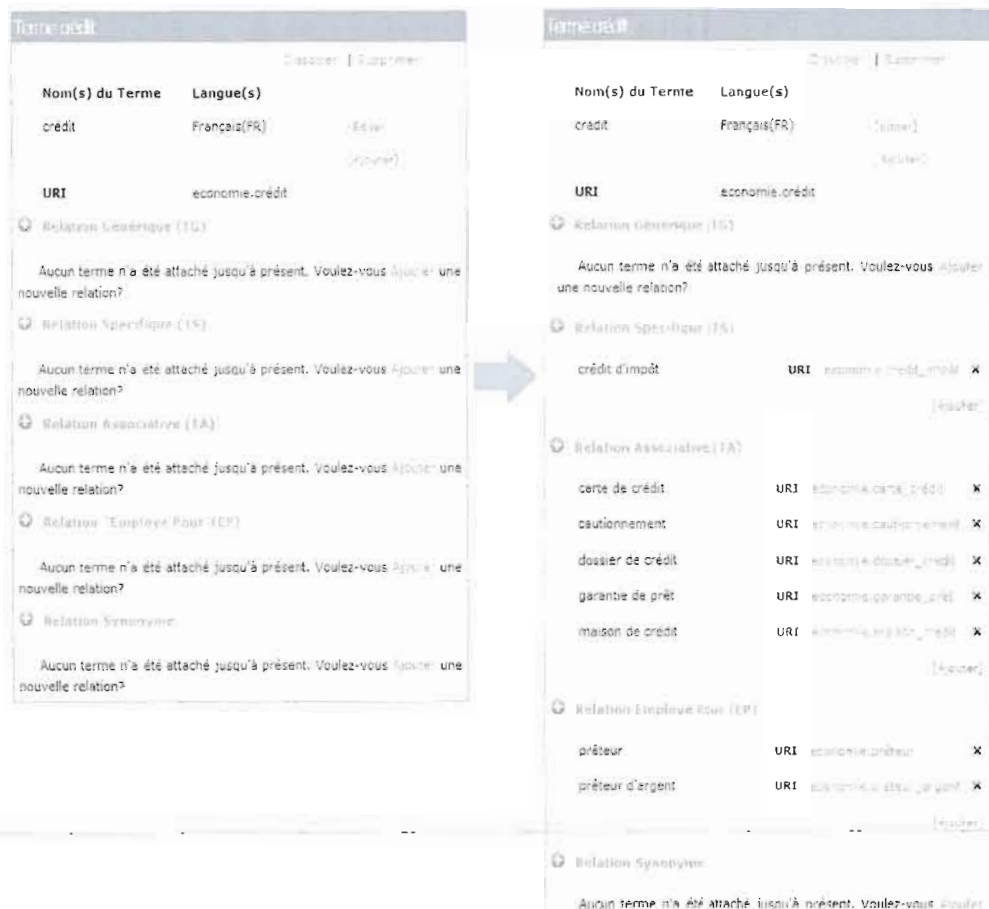


Figure 6.8 - Affichage des propriétés du terme « Crédit » avant et après ajout des relations

Nous réitérons cette opération avec tous les autres termes que nous avons préalablement insérés. Après avoir fini de lier les termes entre eux, nous passons à l'étape suivante de notre scénario; la vérification des informations introduites dans la base de données.

Pour réaliser cette étape, nous avons procédé, tout comme pour les glossaires, à l'implémentation d'un script SQL permettant d'interroger la base de données et de récupérer les informations relatives à un terme à partir des tables concernées.

Le script que nous avons développé est semblable à celui réalisé pour l'étape de vérification pour le glossaire « Banque du Canada ». Ce script est représenté dans la figure ci-après.

PROCESSING SCRIPT...

TABLE -> CONCEPT	TABLE -> CONCEPTSCHEME
- Libelle: crédit	- Libelle: Economie
- URI: economie.credit	- URI: economie
TABLE -> LABELSKOS	
- idLangue: 11	
- libelle : crédit	
- idLabelType: 1	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 103	- id: 1
- conceptCible: 98	- libelle: related
- idSemanticRelation: 1	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 104	- id: 1
- conceptCible: 98	- libelle: related
- idSemanticRelation: 1	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 105	- id: 1
- conceptCible: 98	- libelle: related
- idSemanticRelation: 1	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 106	- id: 1
- conceptCible: 98	- libelle: related
- idSemanticRelation: 1	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 107	- id: 1
- conceptCible: 98	- libelle: related
- idSemanticRelation: 1	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 99	- id: 2
- conceptCible: 98	- libelle: Narrower
- idSemanticRelation: 2	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 101	- id: 10
- conceptCible: 98	- libelle: Close Match
- idSemanticRelation: 10	
TABLE -> SEMANTICRELATION	TABLE -> SEMANTICRELATIONTYPE
- conceptSource: 102	- id: 10
- conceptCible: 98	- libelle: Close Match
- idSemanticRelation: 10	

Figure 6.9 - Résultats de l'exécution du script d'expérimentation du mécanisme de « mapping » sur le nouveau thésaurus

Les résultats obtenus après l'exécution de notre script sont ensuite analysés. Pour réaliser cette analyse, nous commençons d'abord par établir les résultats attendus théoriquement pour après les comparer aux résultats produits par notre script. Le tableau ci-dessous nous résume cette analyse.

Notions	Propriétés SKOS correspondantes	Résultats théorique avec l'exemple de la figure 59	Résultats trouvés dans la BD
Thésaurus	Skos: conceptscheme	Table = conceptscheme - Libelle = 'Economie' - URI = 'économie'	√
Relation	Skos :semantic Relation	Table = 'semanticrelation' - Idconceptsource = 'équivalent à une valeur X ' - idconceptcible = 'équivalent à une valeur Y' (différent pour chaque type de relation) - idsemanticrelationtype = 'équivalent à une valeur A correspondant au type de la relation générique' et/ou - idsemanticrelationtype = 'équivalent à une valeur B correspondant au type de la relation spécifique' et/ou - idsemanticrelationtype = 'équivalent à une valeur C correspondant au type de la relation associative' et/ou - idsemanticrelationtype = 'équivalent à une valeur D correspondant au type de la relation équivalente' et/ou - idsemanticrelationtype = 'équivalent à une valeur E correspondant au type de la relation synonyme'	√
Terme	Skos :Concept	Table = 'concept' - Id = 'équivalent à une valeur X ' - Libelle = 'assouplissement_quantitatif' - URI = 'économie.credit'	√
Libellé d'un terme	Skos :PrefLabel	Table = 'labelskos' - idconcept = 'X' - idLangue = '11' - libelle = 'crédit' - idLabelType = 'valeur du id	√

		correspondant à « PrefLabel » dans la table labeltype	
Relation générique	Skos :broader	Table= 'semanticRelationType' - id= 'A' - libelle = 'borader'	✓
Relation spécifique	Skos :narrower	Table= 'semanticRelationType' - id= 'B' - libelle = 'narrower'	✓
Relation associative	Skos :related	Table= 'semanticRelationType' - id= 'C' - libelle = 'related'	✓
Relation d'équivalence entre termes quasi-synonymes	Skos :closeMatch	Table= 'semanticRelationType' - id= 'D' - libelle = 'closematch'	✓
Relation d'équivalence entre termes synonymes	Skos :exactMatch	Table= 'semanticRelation' - id= 'E' - libelle = 'exactmatch'	✓

Tableau 6.3 – Synthèse de l'analyse entre les résultats d'exécution du script et des résultats théoriques.

Après avoir effectué la vérification des données de la base de données, nous passons à l'étape suivante. Lors de cette étape, nous effectuons plusieurs modifications des éléments d'un terme, entre autre, la suppression d'une relation, l'ajout d'une nouvelle relation et la modification de la langue utilisée pour décrire graphiquement les propriétés d'un terme. Une fois ces modifications apportées, nous réutilisons le même script pour afficher les attributs des tables concernées et comparer par la suite les résultats observés aux modifications réalisées. L'analyse de cette comparaison est tout aussi concluante que celle réalisée lors de l'étape de vérification après insertion de nouveaux termes.

Concernant la dernière étape de notre scénario d'expérimentation, nous avons procédé à la suppression du thésaurus « Économie ». La vérification de la base de données supportant SKOS, nous a permis de constater que tous les termes qu'il contenait ont été supprimés en intégralité. Ceci s'explique par l'emploi de la suppression en cascade.

6.3.3 Conclusion et limites

L'expérimentation de la fonctionnalité de gestion des thésaurus grâce à l'étude de cas du thésaurus de l'activité gouvernementale a été un bon moyen pour tester et valider la capacité de notre outil à gérer et manipuler des thésaurus. La manipulation des données grâce à cette fonctionnalité a été simple et efficace. L'interface que l'outil propose a une prise en main facile et intuitive. Par ailleurs, cette fonctionnalité permet de gérer les thésaurus efficacement.

Il est vrai que l'expérimentation que nous avons produite s'est limitée à un échantillon de quelques dizaines de termes. En effet, lors de la première étape nous n'avons procédé qu'à l'insertion de cet échantillon. Nous nous sommes par ailleurs arrêtés à la modification du seul terme « Crédit » et la vérification de la réussite de la suppression n'a été confirmée que sur la base de quelques identifiants de termes.

Néanmoins, à travers ce petit échantillon, nous avons pu déduire que la fonctionnalité de gestion des thésaurus est apte à répondre aux besoins de l'utilisateur. Les différentes étapes du scénario ont été exécutées avec succès et sans complications. Les données saisies au travers de notre outil ont été sauvegardées correctement dans la base de données de SKOS et aucune perte d'informations n'a été décelée.

6.4. CREATION ET MANIPULATION D'UN SYSTEME DE GESTION DES CONNAISSANCES « KOS »

La gestion de KOS est une des fonctionnalités principales de notre outil. En effet, grâce à cette fonctionnalité, l'outil offre à ses utilisateurs la possibilité de gérer des systèmes de gestion des connaissances autres que les thésaurus et les glossaires qui y sont implémentés par défaut. Pour évaluer et valider cette fonctionnalité, nous procédons à l'étude d'un thésaurus contenant en plus des relations entre les termes, leur définition. Nous nommerons cette sorte de système de gestion des connaissances un « Glosaurus ».

Pour réaliser notre expérimentation, nous recourons au même thésaurus de l'étude de cas précédent, c'est-à-dire, le thésaurus de l'activité gouvernementale qui renferme sur son site Web cette variété de KOS.

6.4.1 Expérimentation

6.4.1.1 Scénario d'expérimentation

Comme dans les sections précédentes, nous établissons avant de commencer notre expérimentation le scénario que nous nous proposons de suivre. Ce scénario sera composé de plusieurs étapes, qui sont les suivantes :

- Création d'un nouveau type de KOS.
- Vérification des informations introduites dans la base de données
- Insertion de termes dans le nouveau KOS
- Vérification du fonctionnement du mécanisme de mapping.

Nous aurions pu ajouter d'autres étapes comme celle de la modification et la suppression d'un terme dans un KOS. Cependant, comme nous avons démontré que les thésaurus et les glossaires, qui sont respectivement des schémas de concepts et des collections, sont gérés efficacement. Par conséquent, quelque soit la nature du KOS, ce dernier ne peut qu'être un schéma de concepts ou une collection et la gestion de l'un comme de l'autre a été vérifiée et validée.

6.4.1.2 Exécution du scénario

Nous entamons notre expérimentation par la création d'un nouveau KOS. Comme nous l'avons précisé un peu plus haut, nous le nommons « Glosaurus ». Etant donné que cet ensemble de données terminologiques repose sur le modèle de thésaurus, nous convenons que l'élément équivalent en SKOS est le schéma de concepts. Tout comme le thésaurus, un « Glosaurus » devrait reprendre les mêmes relations qu'un thésaurus ainsi que des notes permettant d'associer une définition à un terme. La figure qui suit présente l'interface employée pour créer notre nouveau KOS.

Mapping des propriétés

General

Nom du nouveau KOS: Glosaurus

Equivalent du KOS en SKOS: Conceptscheme(Arbre de classification) ▼

Libelle

Est ce qu'au moins un terme du nouveau KOS possède-t-il au moins

un libellé alternatif? ☐ Oui ☒ Non

un libellé erroné? ☐ Oui ☒ Non

un libellé différent des deux précédents? ☐ Oui ☒ Non

Relations

Quel type de relation avez-vous besoin d'utiliser?

Nom de la Relation:	Générique	x
Relation SKOS correspondante	Broader ▼	
Nom de la Relation:	Spécifique	x
Relation SKOS correspondante	Narrower ▼	
Nom de la Relation:	Associative	x
Relation SKOS correspondante	Related ▼	
Nom de la Relation:	Employé Pour	x
Relation SKOS correspondante	Closed Match ▼	
Nom de la Relation:	Synonyme	x
Relation SKOS correspondante	Exact Match ▼	

[Ajouter]

Notes

Quel type de note contient votre nouveau système de gestion des connaissances?

Nom de la Note:	Définition	x
A quelle note SKOS correspondante-elle?	Définition ▼	

[Ajouter]

Figure 6.10 - Processus de création d'un nouveau KOS

Une fois le nouveau KOS produit, nous passons à la vérification de l'insertion des données dans les tables de métadonnées qui sont chargées de stocker les informations relatives aux KOS. Pour vérifier ces informations, nous élaborons un script et l'exécutons. Suite à cette opération, nous obtenons les résultats suivants :

PROCESSING SCRIPT...

TABLE -> METAKOS	
- kosname: glosaurus	
- isCollection: false	
- isConceptScheme: true	
TABLE -> METAKOS_RELATIONS	TABLE -> SEMANTICRELATIONTYPE
- idSkosSemanticRelationType: 3	- Id: 3
- metaKosRelationName : Générique	- semanticRelationType libelle : Broader
TABLE -> METAKOS_RELATIONS	TABLE -> SEMANTICRELATIONTYPE
- idSkosSemanticRelationType: 2	- Id: 2
- metaKosRelationName : Spécifique	- semanticRelationType libelle : Narrower
TABLE -> METAKOS_RELATIONS	TABLE -> SEMANTICRELATIONTYPE
- idSkosSemanticRelationType: 1	- Id: 1
- metaKosRelationName : Associative	- semanticRelationType libelle : related
TABLE -> METAKOS_RELATIONS	TABLE -> SEMANTICRELATIONTYPE
- idSkosSemanticRelationType: 10	- Id: 10
- metaKosRelationName : Employé Pour	- semanticRelationType libelle : Close Match
TABLE -> METAKOS_RELATIONS	TABLE -> SEMANTICRELATIONTYPE
- idSkosSemanticRelationType: 11	- Id: 11
- metaKosRelationName : Synonyme	- semanticRelationType libelle : Exact Match
TABLE -> METAKOS_NOTES	TABLE -> NOTETYPE
- idSkosNote: 1	- id: 1
- metaKosNoteName: Définition	- libelle: Définition

Figure 6.11 - Résultats d'exécution du script de vérification du mécanisme de « mapping » du nouveau KOS

En observant cette figure, nous remarquons que les diverses propriétés introduites lors de l'étape précédente ont été correctement sauvegardées. Ces informations, rappelons le, serviront à générer automatiquement le menu de gestion ainsi que les interfaces permettant de gérer les termes qui le composent.

A présent, nous nous intéressons à la gestion de cet ensemble de données terminologiques. Pour ce faire, nous passons à l'étape suivante de notre scénario. Nous introduisons différents termes depuis le thésaurus de l'activité gouvernementale ainsi que leurs définitions. Afin de réaliser cette étape, nous commençons par la création du « glosaurus » en procédant tel que dans la figure ci-dessous.



Figure 6.12 - Processus de création d'un « Glosaurus »

Après avoir fini de créer le nouveau « glosaurus », nous introduisons quelques termes afin de tester le bon fonctionnement de notre outil. Étant donné que notre outil a réussi à gérer des collections et des schémas de concepts efficacement, il sera normalement apte à bien gérer les « glosaurus » qui sont des schémas de concepts. La figure suivante nous présente l'interface d'ajout d'un nouveau terme.

Figure 6.13 - Interface d'insertion et d'affichage du terme « activité économique » dans un « Glosaurus »

Nous pouvons observer à travers cette figure que les éléments introduits au moment de la création du KOS « glosaurus », ont été repris par cette interface pour sa génération. Ceci démontre d'une manière visuelle la validité de l'outil à générer de nouveaux KOS.

Désormais, nous regardons de près le fonctionnement du mécanisme de mise en correspondance après cette insertion. Pour ce faire, nous développons une fois de plus un script permettant d’afficher les informations introduites lors d’une insertion d’un nouveau terme afin de nous permettre de les analyser par la suite. La figure suivante reflète l’exécution de ce script.

PROCESSING SCRIPT...

```
TABLE -> CONCEPT
- Libelle: cycle économique
- URI: economie.cycle_economique

TABLE -> CONCEPTSCHEME
- Libelle: Economie
- URI: economie_glosaurus

TABLE -> LABELSKOS
- idLangue: 11
- libelle: cycle économique
- idLabelType: 1

TABLE -> SEMANTICRELATION
- conceptSource: 109
- conceptCible: 108
- idSemanticRelation: 1

TABLE -> SEMANTICRELATIONTYPE
- id: 1
- libelle: related

TABLE -> NOTE
- idConcept: 109
- idLangue: 11
- idNotetype: 1
- noteString: Fluctuation périodique plus ou moins régulière de l'act[.]

TABLE -> NOTETYPE
- id: 1
- libelle: Definition

--> END of Experience<--
```

Figure 6.14 - Résultats d'exécution du script d'expérimentation du mécanisme de mapping suite à une insertion du terme « cycle économique »

Nous remarquons à partir de cette figure que le mécanisme de « mapping » des éléments du terme « cycle économique » du « glosaurus » en SKOS, a été accompli tout en respectant les règles identifiées dans le chapitre 4.

6.4.1.3 Conclusion et limites

La fonctionnalité de gestion des KOS a été testée et validée avec succès lors de cette étude de cas. Elle nous a permis de créer, tout d’abord, une nouvelle variété d’ensemble

de données terminologiques avec réussite. Ceci, nous permet aussi de démontrer la capacité de l'outil à créer n'importe quelle sorte de KOS et à gérer efficacement et avec simplicité leurs contenus. En effet, grâce à l'automatisation de la création d'une part du menu permettant l'utilisation d'un nouveau KOS et d'autre part, la génération d'interfaces automatiquement, notre outil permettra à un administrateur de manipuler les termes composant un KOS sans difficulté.

L'expérimentation nous a aussi démontré la facilité avec laquelle un administrateur peut créer un KOS grâce notamment à l'utilisation de questionnaires simples. Cependant, il est à noter, qu'au cours de cette expérimentation, nous nous sommes seulement contentés de la création de termes. Nous avons supposé que les opérations de modification et de suppression fonctionnent bien et ce, parce qu'elles ont été testées et validées lors des expérimentations précédentes concernant les glossaires et les thésaurus, qui ne sont en réalité que des collections et des schémas de concepts.

6.5. CONCLUSION

Après avoir réalisé des expérimentations se basant sur des cas d'études, nous avons pu démontrer et évaluer la capacité de l'outil que nous avons prototypé à répondre aux attentes des utilisateurs. Il est vrai que l'expérimentation que nous avons accomplie est limitée mais cette activité de la méthodologie aurait demandé un effort important dépassant le cadre d'une maîtrise. Néanmoins, nous nous sommes appliqués lors de sa réalisation. En effet, au travers des études de cas, nous avons pu comparer des résultats attendus théoriquement avec ceux fournis par l'application et précisément par l'exploration des tables de la base de données supportant le modèle conceptuel de SKOS. Les résultats auxquels nous sommes parvenus sont concluants. Toutes les principales fonctionnalités qui ont été spécifiées lors du chapitre 4 de ce mémoire fonctionnent correctement. Quant aux fonctionnalités de gestion des ensembles de données terminologiques proposés par défaut par l'outil elles permettent une gestion simple et une manipulation adéquate. Par ailleurs, les données qui sont introduites dans l'outil sont toutes normalisées sous le standard SKOS sans que l'utilisateur ne s'en rende compte. Le mécanisme de mise en correspondance entre les propriétés d'un thésaurus et d'un glossaire, fonctionne convenablement. Malheureusement, nous n'avons pas pu tester au complet une des fonctionnalités de l'outil, celle qui permet la gestion et la création de

KOS par un administrateur. Nous nous sommes arrêtés uniquement à la création d'un seul système de gestion des connaissances, que nous avons proposé par nous-même.

CONCLUSION

L'adoption du standard SKOS par le W3C est un pas vers la normalisation des ensembles de données terminologiques. Désormais, il est possible aux systèmes logiciels de s'échanger des systèmes de gestion des connaissances tels que les glossaires ou thésaurus d'une façon simple et conventionnée et ce, grâce aux éléments composant ce standard. Depuis l'adoption de ce standard, divers outils proposent de manipuler et de gérer des ensembles de données terminologiques basés sur SKOS. Cependant, certains d'entre eux possèdent plusieurs inconvénients. Bon nombre d'outils que nous avons identifiés ne sont pas faciles à utiliser et leur utilisation nécessite parfois une expertise du standard. Ces outils souffrent par ailleurs d'un manque de souplesse quant à la gestion des KOS qui, souvent, ne sont pas conformes aux standards qui les gèrent et ne supportent pas les évolutions de SKOS. Aussi plusieurs outils ne se conforment-ils pas au standard SKOS en n'incluant qu'une partie de ses éléments et en ayant du mal à les faire correspondre aux propriétés des ensembles de données terminologiques.

Au cours de notre travail de recherche, nous nous sommes donc intéressés au développement d'un outil apte à combler ces manques. Pour ce faire, nous avons repris le méta-modèle réalisé lors de travaux précédents et nous avons accompli différentes modifications pour, d'une part convenir avec les évolutions du standard SKOS et d'autre part pour développer notre modèle conceptuel le supportant. Nous avons ensuite procédé à sa validation afin d'implanter la base de données SKOS dans notre outil et nous nous sommes finalement penchés sur la conception et l'implémentation de notre outil logiciel. Ce dernier offre plusieurs fonctionnalités et répond aux exigences du standard SKOS. Il permet non seulement de gérer des glossaires et des thésaurus mais aussi de gérer diverses sortes d'ensembles de données terminologiques. Il offre ainsi une grande flexibilité aux utilisateurs, en leur permettant de créer des KOS de natures différentes tout en normalisant leurs données en SKOS. Aussi, propose-t-il aux utilisateurs d'exporter les ensembles de données terminologiques ainsi que leurs règles de correspondances,

simplifiant en conséquence leurs échanges entre des systèmes hétérogènes. Par ailleurs, notre outil offre à ses utilisateurs des interfaces simples et claires favorisant ainsi son utilisation et élargissant le champ aux utilisateurs qui ne sont pas experts du domaine.

Bien que l'expérimentation que nous avons réalisée démontre cette capacité et soutienne ce que nous avançons, certaines limites sont à signaler et peuvent faire l'objet de travaux futurs. Certes, l'outil est fonctionnel, toutefois des améliorations peuvent y être apportées. Tout d'abord, le questionnaire servant à la construction dans le module de gestion des KOS pourrait être raffiné afin d'améliorer sa compréhension et ainsi être plus précis lors de la construction de KOS. Ensuite, plusieurs contrôles pourraient être rajoutés. Des contrôles au niveau de l'ajout d'un terme, d'une relation ou encore d'une note pourraient être implémentés pour éviter de reproduire des éléments qui existent déjà. Finalement, une fonctionnalité de gestion des utilisateurs serait intéressante pour notre outil, tout comme une fonctionnalité d'importation de nouveaux KOS normalisés en SKOS. Cette dernière pourrait faire l'objet d'un travail de recherche approfondi perfectionnant l'outil, en conséquence. Identifier et implémenter une approche établissant les correspondances entre les éléments importés et le mécanisme de correspondance de l'outil, rendraient notre outil plus consistant, plus performant et plus complet.

BIBLIOGRAPHIE

- Archer L.B. 1964. *Systematic method for designers*. In N. Cross (Ed.), *Developments in design methodology*, p.57-82.
- Alistair Miles et Sean Bechhofer. 2009. *SKOS Simple Knowledge Organization System Reference*. Recommendation. World Wide Web Consortium.
- Alistair Miles, Mark van Assem, Véronique Malaisé. 2006. *A method to convert thesauri to SKOS*, Springer Berlin / Heidelberg, vol. 4011/2006, p.95-109.
- Antoine Isaac et Ed Summe. 2008. *SKOS Simple Knowledge Organization System Primer*. World Wide Web Consortium.
- Anne Tasso et Sébastien Ermacore, 2004. *Initiation à JSP*, Edition Eyrolles.
- Banque du Canada, 2010. <http://www.banqueducanada.ca/> (Page consultée en 2010)
- Dan Brickley, Libby Miller et Kate Sharp. 2004. Semantic Web Standardization in Europe: SWAD-Europe, ERCIM News, No 59, October 2004.
- Derek Austin et Janet Waters. 1980. *Principes Directeurs pour l'Établissement et le Développement des Thésaurus Multilingues*. Bibliographic Systems and Standards Office, The British Library. p.8.
- Godfray et H Charles J. 2002. *Challenges for taxonomy*. Nature, vol.417, no. 6884, p.17-19. Macmillan Magazines.
- Hevener A.R, March S.T., et Park J. 2004. *Design Research in Information Systems Research*. MIS Quarterly, vol.28, p.75-105.
- Jarvinen P. 2007. *Action Research is Similar to Design Science*. Quality & Quantity, vol.41(1), p.37-54.

- Javier Lacasta, Javier Nogueras-Iso, Francisco Javier Lopez-Pellicer, Pedro Rafael Muro-Medrano et Francisco Javier Zarazga-Soria. 2007. *ThManager: An OpenSource Tool for Creating and Visualizing SKOS*. Information Technology and Libraries, vol. 26(3), p. 39–51.
- Jouni Tuominen, Matias Frosterus, Kim Viljanen et Eero Hyvönen. 2009. *ONKI SKOS Server for publishing and utilizing SKOS vocabularies and ontologies as services*. Springer Berlin / Heidelberg, vol. 5554/2009, p. 768-780.
- Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger et Samir Chatterjee. 2007. *A Design Research Methodology for Information Systems Research*. Journal of Management Information Systems, vol. 24, no 3, p. 45-78.
- Laurent Audibert. 2009. *UML 2 de l'apprentissage à la pratique : Présentation des diagrammes UML-Language de contraintes OCL-Introduction aux patrons de conception-Mises en œuvre d'UML*. Ellipses Marketing.
- M Zacklad. 2007, *Classification, Thésaurus, Ontologies, Folksonomies: comparaison de point de vue de la recherche ouverte d'information (ROI)*, 35ème congrès annuel de l'association Canadienne des sciences de l'information.
- Nunamaker, J.F., Chen, M., and Purdin. 1991. *Systems Development in Information Systems Research*. Journal of Management Information Systems, 7, 3, 89-106.
- Observatoire international des coopératives de services financiers, 2010. <http://observatoire.coopfinance.hec.ca/login.html> (Page consultée en 2010)
- Oracle. 2010. *MySQL 5.5 Reference Manual*. <http://dev.mysql.com/doc/refman/5.5/en/> (Page consultée en 2010)
- Peter H. Raven, Brent Berlin et Dennis E. Breedlove. 1971. *The Origins of Taxonomy*. Science, vol. 174, no. 4015, 17 December 1971, pp1210-1213.
- R Lega. 1998. *Production, maintenance and visualization of multilingual thesaurus through THESmain and THESshow*, IFLA Journal, Vol.24, p. 130-131.

- Semantic Web Deployment Working Group. 2006. <http://www.w3.org/2006/07/SWD/>. World Wide Web Consortium. (page consultée 2010)
- Simon Jupp, Sean Bechhofer et Robert Stevens. 2008. *A Flexible API and Editor for SKOS*. European Semantic Web Symposium, p. 506-520.
- Sean Bechhofer, Frank Van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider et Lynn Andrea Stein. 2004. *OWL Web Ontology Language Reference*. Recommendation. World Wide Web Consortium.
- Sun Microsystems. 2010. *Documentation, Training & Support*. <http://netbeans.org/kb/index.html> (page consultée en 2010)
- T. Bray, J. Paoli. 1998. *Extensible Markup Language (XML) 1.0*. Recommendation. World Wide Web Consortium.
- Thomas Gruber, Ling Liu et M. Tamer Özsu (Eds.). 2008. *Ontology*. Encyclopedia of Database Systems. Springer-Verlag.
- T. Gruber. 1994. *Toward principles for the design of ontologies used for knowledge sharing*. International Journal of Human-Computer Studies, Vol. 43, p. 907-928.
- T. Gruber. 1993. *A translation approach to portable ontology specifications*, Knowledge Acquisition, Vol. 5, p.199-220.
- Tim Berners-Lee, James Hendler et Ora Lassila. 2001. *The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*. Scientific American, 284, no 5, p. 34-43.
- Wolf-Dieter Batschi, Bruno Felluga, Rudolf Legat, Paolo Plini, Hermann Stallbaumer et Konrad L. Zirm. 2002. *SuperThes : A new software for Construction, Maintenance and Visualisation of Multilingual Thesauri*.

World Wide Web Consortium. 2001. « About W3C », W3C (Page consultée en 2009).

Wright Sue Ellen et Budin Gerhard, 1997. *Handbook of Terminology Management, Vol 1*. Amsterdam/Philadelphia: John Benjamins Publishing Company.